



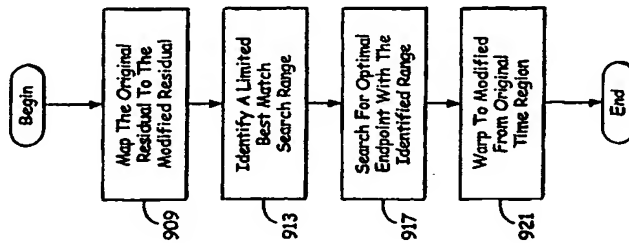
## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>7</sup> : G10L 19/08, 19/12	(11) International Publication Number: WO 00/11653
(21) International Application Number: PCT/US99/19175	(43) International Publication Date: 2 March 2000 (02.03.00)
(22) International Filing Date: 24 August 1999 (24.08.99)	(81) Designated States: CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(30) Priority Date: 24 August 1998 (24.08.98) US 09/154,675 18 September 1998 (18.09.98) US	Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.
(71) Applicant: CONEXANT SYSTEMS, INC. [US/US]; 4311 Jamboree Road, Newport Beach, CA 92660-3095 (US).	
(72) Inventor: GAO, Yang; 26586 Sun Torini Road, Mission Viejo, CA 92692-6101 (US).	
(74) Agent: BENNETT, James, D.; Akin, Gump, Strauss, Haber & Feld, L.L.P., Suite 1900, 816 Congress Avenue, Austin, TX 78701 (US).	

## (54) Title: SPEECHENCODER USING CONTINUOUS WARPING COMBINED WITH LONG TERM PREDICTION

## (57) Abstract

A multi-rate speech codec supports a plurality of encoding bit rate modes by adaptively selecting encoding bit rate modes to match communication channel restrictions. In higher bit rate encoding modes, an accurate representation of speech through CELP (code excited linear prediction) and other associated modeling parameters are generated for higher quality decoding and reproduction. To support lower bit rate encoding modes, a variety of techniques are applied many of which involve the classification of the input signal. The speech encoder continuously warps a weighted speech signal in long term preprocessing. The continuous warping is applied to a linear pitch lag contour that enables fast searching through linear time weighting. Optimal searching is performed within a limited range that is defined at least in part on sharpness and speech classification. The speech encoder generates the linear pitch lag contour from previous and current pitch lag values. Such continuous warping may also be applied in an open loop approach to the residual signal.



## FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT:

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Togo
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GR	Greece	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MY	Malaysia	UZ	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon	KR	Republic of Korea	PL	Poland		
CN	China	KZ	Kazakhstan	PT	Portugal		
CU	Cuba	LC	Liechtenstein	RO	Romania		
CZ	Czech Republic	LI	Liechtenstein	RU	Russian Federation		
DE	Germany	LR	Liberia	SD	Sudan		
DK	Denmark			SE	Sweden		
EE	Estonia			SG	Singapore		

SPEECH ENCODER USING CONTINUOUS WARPING COMBINED WITH LONG TERM PREDICTION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

TITLE: SPEECH ENCODER USING CONTINUOUS WARPING  
IN LONG TERM PREPROCESSING

TITLE

SPEECH ENCODER USING CONTINUOUS WARPING  
IN LONG TERM PREPROCESSING

SPECIFICATION

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application is based on U.S. Patent Application Ser. No. 09/154,675, filed September 18, 1998. This application is based on U.S. Provisional Application Serial No. 60/097,569, filed on August 24, 1998. All of such applications are hereby incorporated herein by reference in their entirety and made part of the present application.

INCORPORATION BY REFERENCE

The following applications are hereby incorporated herein by reference in their entirety and made part of the present application:

- 1) U.S. Provisional Application Serial No. 60/097,569 (Attorney Docket No. 98RSS325), filed August 24, 1998;
- 2) U.S. Patent Application Serial No. 09/154,675 (Attorney Docket No. 97RSS383), filed September 18, 1998;
- 3) U.S. Patent Application Serial No. 09/156,814 (Attorney Docket No. 98RSS365), filed September 18, 1998;
- 4) U.S. Patent Application Serial No. 09/156,649 (Attorney Docket No. 95E020), filed September 18, 1998;
- 5) U.S. Patent Application Serial No. 09/156,648 (Attorney Docket No. 98RSS228), filed September 18, 1998;
- 6) U.S. Patent Application Serial No. 09/156,650 (Attorney Docket No. 98RSS343), filed September 18, 1998;
- 7) U.S. Patent Application Serial No. 09/156,832 (Attorney Docket No. 97RSS039), filed September 18, 1998.

- 8) U.S. Patent Application Serial No. 09/154,654 (Attorney Docket No. 98RSS344), filed September 18, 1998;
- 9) U.S. Patent Application Serial No. 09/154,657 (Attorney Docket No. 98RSS328), filed September 18, 1998;
- 10) U.S. Patent Application Serial No. 09/156,826 (Attorney Docket No. 98RSS382), filed September 18, 1998;
- 11) U.S. Patent Application Serial No. 09/154,662 (Attorney Docket No. 98RSS383), filed September 18, 1998;
- 12) U.S. Patent Application Serial No. 09/154,653 (Attorney Docket No. 98RSS406), filed September 18, 1998;
- 13) U.S. Patent Application Serial No. 09/154,660 (Attorney Docket No. 98RSS384), filed September 18, 1998;
- 14) U.S. Patent Application Serial No. 09/198,414 (Attorney Docket No. 97RSS039CIP), filed November 24, 1998.

## BACKGROUND

### 1. Technical Field

The present invention relates generally to speech encoding and decoding in voice communication systems; and, more particularly, it relates to various techniques used with code-excited linear prediction coding to obtain high quality speech reproduction through a limited bit rate communication channel.

### 2. Related Art

Signal modeling and parameter estimation play significant roles in communicating voice information with limited bandwidth constraints. To model basic speech sounds, speech signals are sampled as a discrete waveform to be digitally processed. In one type of signal coding technique called LPC (linear predictive coding), the signal value at any particular time index is modeled as a linear function of previous values. A subsequent signal is thus linearly predictable according to an earlier value. As a result, efficient signal representations can be determined by estimating and applying certain prediction parameters to represent the signal.

Applying LPC techniques, a conventional source encoder operates on speech signals to extract modeling and parameter information for communication to a conventional source decoder via a communication channel. Once received, the decoder attempts to reconstruct a counterpart signal for playback that sounds to a human ear like the original speech.

A certain amount of communication channel bandwidth is required to communicate the modeling and parameter information to the decoder. In embodiments, for example where the channel bandwidth is shared and real-time reconstruction is necessary, a reduction in the required bandwidth proves beneficial. However, using conventional modeling techniques, the quality

requirements in the reproduced speech limit the reduction of such bandwidth below certain levels.

In conventional coding systems employing long term preprocessing, a modified residual is produced as a new reference for current excitation. The goal is to produce a modified residual that better matches a coded pitch contour (or delay contour) than the original residual so that the LTP gain is higher. This is attempted in conventional systems by individually shifting the pitch pulses to match the pitch contour, requiring reliable endpoint detection of a segment to be shifted to maintain signal continuity. Using such an open loop approach with pulse shifting results in quality problems in speech reproduction.

Additionally, in using such and other conventional approaches, the amount of pitch lag information that must be transmitted is relatively large in view of the limitations often placed on the channel bit rate. For example, 8 bits might be required to encode pitch lag for a first subframe (of 5ms duration) followed perhaps by 5 bits for pitch lag changes in a second subframe, resulting in a relatively large amount of bandwidth allocation, e.g., 1.3 kbps (kilobits per second), just for the pitch lag information.

Further limitations and disadvantages of conventional systems will become apparent to one of skill in the art after reviewing the remainder of the present application with reference to the drawings.

### SUMMARY OF THE INVENTION

Various aspects of the present invention can be found in an embodiment of a speech encoder that uses long term preprocessing of a speech signal wherein the speech signal has a previous pitch lag and a current pitch lag. Therein, the speech encoder comprises an adaptive codebook and an encoder processing circuit coupled to the adaptive codebook. Using estimates of the previous pitch lag and the current pitch lag, the encoder processing circuit generates a pitch lag contour. The encoder processing circuit continuously warps the speech signal to the pitch lag contour.

Many possible variations and further aspects of such a speech encoder are possible. For example, the speech signal may comprise either a weighted speech signal or a residual signal. The pitch lag contour may comprise a linear segment bounded by the estimates of the previous pitch lag and the current pitch lag, and continuous warping may involve warping the speech signal from a first time region to a second time region. Additionally, for example, the encoder processing circuit may search for a best local delay using linear time weighting, and/or perform the estimation of the current pitch lag.

Further aspects of the present invention may be found in an alternate embodiment of a speech encoder that uses long term preprocessing of a speech signal having a pitch lag. As before, the speech encoder comprises an adaptive codebook and an encoder processing circuit coupled thereto. The encoder processing circuit estimates the pitch lag, and, based on such estimate, applies continuous warping of the speech signal.

Other variations and further aspects such as those mentioned previously also apply to this embodiment. For example, the speech signal might comprise a weighted speech signal or a residual signal. The encoder processing circuit may search for a best local delay using linear



time weighting, or conduct continuous warping by translating the speech signal from a first time region to a second time region.

Other aspects, advantages and novel features of the present invention will become apparent from the following detailed description of the invention when considered in conjunction with the accompanying drawings.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

Fig. 1a is a schematic block diagram of a speech communication system illustrating the use of source encoding and decoding in accordance with the present invention.

Fig. 1b is a schematic block diagram illustrating an exemplary communication device utilizing the source encoding and decoding functionality of Fig. 1a.

Figs. 2-4 are functional block diagrams illustrating a multi-step encoding approach used by one embodiment of the speech encoder illustrated in Figs. 1a and 1b. In particular, Fig. 2 is a functional block diagram illustrating of a first stage of operations performed by one embodiment of the speech encoder of Figs. 1a and 1b. Fig. 3 is a functional block diagram of a second stage of operations, while Fig. 4 illustrates a third stage.

Fig. 5 is a block diagram of one embodiment of the speech decoder shown in Figs. 1a and 1b having corresponding functionality to that illustrated in Figs. 2-4.

Fig. 6 is a block diagram of an alternate embodiment of a speech encoder that is built in accordance with the present invention.

Fig. 7 is a block diagram of an embodiment of a speech decoder having corresponding functionality to that of the speech encoder of Fig. 6.

Fig. 8a is a timing diagram of an exemplary pitch lag contour over two speech frames to which continuous warping techniques are applied in accordance with the present invention.

Fig. 8b is a timing diagram illustrating a linear pitch contour to which continuous warping of the original pitch lag contour is applied in accordance with the present invention.

Fig. 8c is a timing diagram illustrating the use of the linear pitch lag contour of Fig. 8b which can be represented by a lesser number of bits than the original pitch lag contour of Fig. 8a.

Fig. 9 is a flow diagram illustrating an embodiment of the continuous warping approach and an associated fast searching process used by an encoder of the present invention to carry out the functionality described in reference to Figs. 8a-c on a residual signal using an open loop approach.

Fig. 10 is a flow diagram illustrating an alternate embodiment of functionality of a speech encoder of the present invention that performs continuous warping to the weighted speech signal in a closed loop approach.

#### DETAILED DESCRIPTION

Fig. 1a is a schematic block diagram of a speech communication system illustrating the use of source encoding and decoding in accordance with the present invention. Therein, a speech communication system 100 supports communication and reproduction of speech across a communication channel 103. Although it may comprise for example a wire, fiber or optical link, the communication channel 103 typically comprises, at least in part, a radio frequency link that often must support multiple, simultaneous speech exchanges requiring shared bandwidth resources such as may be found with cellular telephony embodiments.

Although not shown, a storage device may be coupled to the communication channel 103 to temporarily store speech information for delayed reproduction or playback, e.g., to perform answering machine functionality, voiced email, etc. Likewise, the communication channel 103 might be replaced by such a storage device in a single device embodiment of the communication system 100 that, for example, merely records and stores speech for subsequent playback.

In particular, a microphone 111 produces a speech signal in real time. The microphone 111 delivers the speech signal to an A/D (analog to digital) converter 115. The A/D converter 115 converts the speech signal to a digital form then delivers the digitized speech signal to a speech encoder 117.

The speech encoder 117 encodes the digitized speech by using a selected one of a plurality of encoding modes. Each of the plurality of encoding modes utilizes particular techniques that attempt to optimize quality of resultant reproduced speech. While operating in any of the plurality of modes, the speech encoder 117 produces a series of modeling and parameter information (hereinafter "speech indices"), and delivers the speech indices to a channel encoder 119.

The channel encoder 119 coordinates with a channel decoder 131 to deliver the speech indices across the communication channel 103. The channel decoder 131 forwards the speech indices to a speech decoder 133. While operating in a mode that corresponds to that of the speech encoder 117, the speech decoder 133 attempts to recreate the original speech from the speech indices as accurately as possible at a speaker 137 via a D/A (digital to analog) converter 135.

The speech encoder 117 adaptively selects one of the plurality of operating modes based on the data rate restrictions through the communication channel 103. The communication channel 103 comprises a bandwidth allocation between the channel encoder 119 and the channel decoder 131. The allocation is established, for example, by telephone switching networks wherein many such channels are allocated and reallocated as need arises. In one such embodiment, either a 22.8 kbps (kilobits per second) channel bandwidth, i.e., a full rate channel, or a 11.4 kbps channel bandwidth, i.e., a half rate channel, may be allocated.

With the full rate channel bandwidth allocation, the speech encoder 117 may adaptively select an encoding mode that supports a bit rate of 11.0, 8.0, 6.65 or 5.8 kbps. The speech encoder 117 adaptively selects an either 8.0, 6.65, 5.8 or 4.5 kbps encoding bit rate mode when only the half rate channel has been allocated. Of course these encoding bit rates and the aforementioned channel allocations are only representative of the present embodiment. Other variations to meet the goals of alternate embodiments are contemplated.

With either the full or half rate allocation, the speech encoder 117 attempts to communicate using the highest encoding bit rate mode that the allocated channel will support. If the allocated channel is or becomes noisy or otherwise restrictive to the highest or higher encoding bit rates, the speech encoder 117 adapts by selecting a lower bit rate encoding mode.

Similarly, when the communication channel 103 becomes more favorable, the speech encoder 117 adapts by switching to a higher bit rate encoding mode.

With lower bit rate encoding, the speech encoder 117 incorporates various techniques to generate better low bit rate speech reproduction. Many of the techniques applied are based on characteristics of the speech itself. For example, with lower bit rate encoding, the speech encoder 117 classifies noise, unvoiced speech, and voiced speech so that an appropriate modeling scheme corresponding to a particular classification can be selected and implemented. Thus, the speech encoder 117 adaptively selects from among a plurality of modeling schemes those most suited for the current speech. The speech encoder 117 also applies various other techniques to optimize the modeling as set forth in more detail below.

Fig. 1b is a schematic block diagram illustrating several variations of an exemplary communication device employing the functionality of Fig. 1a. A communication device 151 comprises both a speech encoder and decoder for simultaneous capture and reproduction of speech. Typically within a single housing, the communication device 151 might, for example, comprise a cellular telephone, portable telephone, computing system, etc. Alternatively, with some modification to include for example a memory element to store encoded speech information the communication device 151 might comprise an answering machine, a recorder, voice mail system, etc.

A microphone 155 and an A/D converter 157 coordinate to deliver a digital voice signal to an encoding system 159. The encoding system 159 performs speech and channel encoding and delivers resultant speech information to the channel. The delivered speech information may be destined for another communication device (not shown) at a remote location.

As speech information is received, a decoding system 165 performs channel and speech decoding then coordinates with a D/A converter 167 and a speaker 169 to reproduce something that sounds like the originally captured speech.

The encoding system 159 comprises both a speech processing circuit 185 that performs speech encoding, and a channel processing circuit 187 that performs channel encoding. Similarly, the decoding system 165 comprises a speech processing circuit 189 that performs speech decoding, and a channel processing circuit 191 that performs channel decoding.

Although the speech processing circuit 185 and the channel processing circuit 187 are separately illustrated, they might be combined in part or in total into a single unit. For example, the speech processing circuit 185 and the channel processing circuit 187 might share a single DSP (digital signal processor) and/or other processing circuitry. Similarly, the speech processing circuit 189 and the channel processing circuit 191 might be entirely separate or combined in part or in whole. Moreover, combinations in whole or in part might be applied to the speech processing circuits 185 and 189, the channel processing circuits 187 and 191, the processing circuits 185, 187, 189 and 191, or otherwise.

The encoding system 159 and the decoding system 165 both utilize a memory 161. The speech processing circuit 185 utilizes a fixed codebook 181 and an adaptive codebook 183 of a speech memory 177 in the source encoding process. The channel processing circuit 187 utilizes a channel memory 175 to perform channel encoding. Similarly, the speech processing circuit 189 utilizes the fixed codebook 181 and the adaptive codebook 183 in the source decoding process. The channel processing circuit 187 utilizes the channel memory 175 to perform channel decoding.

Although the speech memory 177 is shared as illustrated, separate copies thereof can be assigned for the processing circuits 185 and 189. Likewise, separate channel memory can be allocated to both the processing circuits 187 and 191. The memory 161 also contains software utilized by the processing circuits 185, 187, 189 and 191 to perform various functionality required in the source and channel encoding and decoding processes.

Figs. 2-4 are functional block diagrams illustrating a multi-step encoding approach used by one embodiment of the speech encoder illustrated in Figs. 1a and 1b. In particular, Fig. 2 is a functional block diagram illustrating of a first stage of operations performed by one embodiment of the speech encoder shown in Figs. 1a and 1b. The speech encoder, which comprises encoder processing circuitry, typically operates pursuant to software instruction carrying out the following functionality.

At a block 215, source encoder processing circuitry performs high pass filtering of a speech signal 211. The filter uses a cutoff frequency of around 80 Hz to remove, for example, 60 Hz power line noise and other lower frequency signals. After such filtering, the source encoder processing circuitry applies a perceptual weighting filter as represented by a block 219. The perceptual weighting filter operates to emphasize the valley areas of the filtered speech signal.

If the encoder processing circuitry selects operation in a pitch preprocessing (PP) mode as indicated at a control block 245, a pitch preprocessing operation is performed on the weighted speech signal at a block 225. The pitch preprocessing operation involves warping the weighted speech signal to match interpolated pitch values that will be generated by the decoder processing circuitry. When pitch preprocessing is applied, the warped speech signal is designated a first target signal 229. If pitch preprocessing is not selected the control block 245, the weighted

speech signal passes through the block 225 without pitch preprocessing and is designated the first target signal 229.

As represented by a block 255, the encoder processing circuitry applies a process wherein a contribution from an adaptive codebook 257 is selected along with a corresponding gain 257 which minimize a first error signal 253. The first error signal 253 comprises the difference between the first target signal 229 and a weighted, synthesized contribution from the adaptive codebook 257.

At blocks 247, 249 and 251, the resultant excitation vector is applied after adaptive gain reduction to both a synthesis and a weighting filter to generate a modeled signal that best matches the first target signal 229. The encoder processing circuitry uses LPC (linear predictive coding) analysis, as indicated by a block 239, to generate filter parameters for the synthesis and weighting filters. The weighting filters 219 and 251 are equivalent in functionality.

Next, the encoder processing circuitry designates the first error signal 253 as a second target signal for matching using contributions from a fixed codebook 261. The encoder processing circuitry searches through at least one of the plurality of subcodebooks within the fixed codebook 261 in an attempt to select a most appropriate contribution while generally attempting to match the second target signal.

More specifically, the encoder processing circuitry selects an excitation vector, its corresponding subcodebook and gain based on a variety of factors. For example, the encoding bit rate, the degree of minimization, and characteristics of the speech itself as represented by a block 279 are considered by the encoder processing circuitry at control block 275. Although many other factors may be considered, exemplary characteristics include speech classification, noise level, sharpness, periodicity, etc. Thus, by considering other such factors, a first

subcodebook with its best excitation vector may be selected rather than a second subcodebook's best excitation vector even though the second subcodebook's better minimizes the second target signal 265.

Fig. 3 is a functional block diagram depicting of a second stage of operations performed by the embodiment of the speech encoder illustrated in Fig. 2. In the second stage, the speech encoding circuitry simultaneously uses both the adaptive the fixed codebook vectors found in the first stage of operations to minimize a third error signal 311.

The speech encoding circuitry searches for optimum gain values for the previously identified excitation vectors (in the first stage) from both the adaptive and fixed codebooks 257 and 261. As indicated by blocks 307 and 309, the speech encoding circuitry identifies the optimum gain by generating a synthesized and weighted signal, i.e., via a block 301 and 303, that best matches the first target signal 229 (which minimizes the third error signal 311). Of course if processing capabilities permit, the first and second stages could be combined wherein joint optimization of both gain and adaptive and fixed codebook vector selection could be used.

Fig. 4 is a functional block diagram depicting of a third stage of operations performed by the embodiment of the speech encoder illustrated in Figs. 2 and 3. The encoder processing circuitry applies gain normalization, smoothing and quantization, as represented by blocks 401, 403 and 405, respectively, to the jointly optimized gains identified in the second stage of encoder processing. Again, the adaptive and fixed codebook vectors used are those identified in the first stage processing.

With normalization, smoothing and quantization functionally applied, the encoder processing circuitry has completed the modeling process. Therefore, the modeling parameters identified are communicated to the decoder. In particular, the encoder processing circuitry

delivers an index to the selected adaptive codebook vector to the channel encoder via a multiplexor 419. Similarly, the encoder processing circuitry delivers the index to the selected fixed codebook vector, resultant gains, synthesis filter parameters, etc., to the multiplexor 419. The multiplexor 419 generates a bit stream 421 of such information for delivery to the channel encoder for communication to the channel and speech decoder of receiving device.

Fig. 5 is a block diagram of an embodiment illustrating functionality of speech decoder having corresponding functionality to that illustrated in Figs. 2-4. As with the speech encoder, the speech decoder, which comprises decoder processing circuitry, typically operates pursuant to software instruction carrying out the following functionality.

A demultiplexor 511 receives a bit stream 513 of speech modeling indices from an often remote encoder via a channel decoder. As previously discussed, the encoder selected each index value during the multi-stage encoding process described above in reference to Figs. 2-4. The decoder processing circuitry utilizes indices, for example, to select excitation vectors from an adaptive codebook 515 and a fixed codebook 519, set the adaptive and fixed codebook gains at a block 521, and set the parameters for a synthesis filter 531.

With such parameters and vectors selected or set, the decoder processing circuitry generates a reproduced speech signal 539. In particular, the codebooks 515 and 519 generate excitation vectors identified by the indices from the demultiplexor 511. The decoder processing circuitry applies the indexed gains at the block 521 to the vectors which are summed. At a block 527, the decoder processing circuitry modifies the gains to emphasize the contribution of vector from the adaptive codebook 515. At a block 529, adaptive tilt compensation is applied to the combined vectors with a goal of flattening the excitation spectrum. The decoder processing circuitry performs synthesis filtering at the block 531 using the flattened excitation signal.

Finally, to generate the reproduced speech signal 539, post filtering is applied at a block 535 deemphasizing the valley areas of the reproduced speech signal 539 to reduce the effect of distortion.

In the exemplary cellular telephony embodiment of the present invention, the A/D converter 115 (Fig. 1a) will generally involve analog to uniform digital PCM including: 1) an input level adjustment device; 2) an input anti-aliasing filter; 3) a sample-and-hold device sampling at 8 kHz; and 4) analog to uniform digital conversion to 13-bit representation.

Similarly, the D/A converter 135 will generally involve uniform digital PCM to analog including: 1) conversion from 13-bit/8 kHz uniform PCM to analog; 2) a hold device; 3) reconstruction filter including  $x/\sin(x)$  correction; and 4) an output level adjustment device.

In terminal equipment, the A/D function may be achieved by direct conversion to 13-bit uniform PCM format, or by conversion to 8-bit/A-law compounded format. For the D/A operation, the inverse operations take place.

The encoder 117 receives data samples with a resolution of 13 bits left justified in a 16-bit word. The three least significant bits are set to zero. The decoder 133 outputs data in the same format. Outside the speech codec, further processing can be applied to accommodate traffic data having a different representation.

A specific embodiment of an AMR (adaptive multi-rate) codec with the operational functionality illustrated in Figs. 2-5 uses five source codecs with bit-rates 11.0, 8.0, 6.65, 5.8 and 4.55 kbps. Four of the highest source coding bit-rates are used in the full rate channel and the four lowest bit-rates in the half rate channel.

All five source codecs within the AMR codec are generally based on a code-excited linear predictive (CELP) coding model. A 10th order linear prediction (LP), or short-term,

synthesis filter, e.g., used at the blocks 249, 267, 301, 407 and 531 (of Figs. 2-5), is used which is given by:

$$H(z) = \frac{1}{B(z)} = \frac{1}{1 + \sum_{i=1}^m \hat{a}_i z^{-i}}, \quad (1)$$

where  $\hat{a}_i, i = 1, \dots, m$ , are the (quantized) linear prediction (LP) parameters.

A long-term filter, i.e., the pitch synthesis filter, is implemented using the either an adaptive codebook approach or a pitch pre-processing approach. The pitch synthesis filter is given by:

$$\frac{1}{B(z)} = \frac{1}{1 - \hat{g}_p z^{-T}}, \quad (2)$$

where  $T$  is the pitch delay and  $\hat{g}_p$  is the pitch gain.

With reference to Fig. 2, the excitation signal at the input of the short-term LP synthesis filter at the block 249 is constructed by adding two excitation vectors from the adaptive and the fixed codebooks 257 and 261, respectively. The speech is synthesized by feeding the two properly chosen vectors from these codebooks through the short-term synthesis filter at the block 249 and 267, respectively.

The optimum excitation sequence in a codebook is chosen using an analysis-by-synthesis search procedure in which the error between the original and synthesized speech is minimized according to a perceptually weighted distortion measure. The perceptual weighting filter, e.g., at the blocks 251 and 268, used in the analysis-by-synthesis search technique is given by:

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)}, \quad (3)$$

where  $A(z)$  is the unquantized LP filter and  $0 < \gamma_2 < \gamma_1 \leq 1$  are the perceptual weighting factors. The values  $\gamma_1 = [0.9, 0.94]$  and  $\gamma_2 = 0.6$  are used. The weighting filter, e.g., at the

blocks 251 and 268, uses the unquantized LP parameters while the formant synthesis filter, e.g., at the blocks 249 and 267, uses the quantized LP parameters. Both the unquantized and quantized LP parameters are generated at the block 239.

The present encoder embodiment operates on 20 ms (millisecond) speech frames corresponding to 160 samples at the sampling frequency of 8000 samples per second. At each 160 speech samples, the speech signal is analyzed to extract the parameters of the CELP model, i.e., the LP filter coefficients, adaptive and fixed codebook indices and gains. These parameters are encoded and transmitted. At the decoder, these parameters are decoded and speech is synthesized by filtering the reconstructed excitation signal through the LP synthesis filter.

More specifically, LP analysis at the block 239 is performed twice per frame but only a single set of LP parameters is converted to line spectrum frequencies (LSF) and vector quantized using predictive multi-stage quantization (PMVQ). The speech frame is divided into subframes. Parameters from the adaptive and fixed codebooks 257 and 261 are transmitted every subframe. The quantized and unquantized LP parameters or their interpolated versions are used depending on the subframe. An open-loop pitch lag is estimated at the block 241 once or twice per frame for PP mode or LTP mode, respectively.

Each subframe, at least the following operations are repeated. First, the encoder processing circuitry (operating pursuant to software instruction) computes  $x(n)$ , the first target signal 229, by filtering the LP residual through the weighted synthesis filter  $W(z)H(z)$  with the initial states of the filters having been updated by filtering the error between LP residual and excitation. This is equivalent to an alternate approach of subtracting the zero input response of the weighted synthesis filter from the weighted speech signal.

Second, the encoder processing circuitry computes the impulse response,  $h(n)$ , of the weighted synthesis filter. Third, in the LTP mode, closed-loop pitch analysis is performed to find the pitch lag and gain, using the first target signal 229,  $x'(n)$ , and impulse response,  $h(n)$ , by searching around the open-loop pitch lag. Fractional pitch with various sample resolutions are used.

In the PP mode, the input original signal has been pitch-preprocessed to match the interpolated pitch contour, so no closed-loop search is needed. The LTP excitation vector is computed using the interpolated pitch contour and the past synthesized excitation.

Fourth, the encoder processing circuitry generates a new target signal  $x_1(n)$ , the second target signal 253, by removing the adaptive codebook contribution (filtered adaptive code vector) from  $x'(n)$ . The encoder processing circuitry uses the second target signal 253 in the fixed codebook search to find the optimum innovation.

Fifth, for the 11.0 kbps bit rate mode, the gains of the adaptive and fixed codebook are scalar quantized with 4 and 5 bits respectively (with moving average prediction applied to the fixed codebook gain). For the other modes the gains of the adaptive and fixed codebook are vector quantized (with moving average prediction applied to the fixed codebook gain).

Finally, the filter memories are updated using the determined excitation signal for finding the first target signal in the next subframe.

The bit allocation of the AMR codec modes is shown in table 1. For example, for each 20 ms speech frame, 220, 160, 133, 116 or 91 bits are produced, corresponding to bit rates of 11.0, 8.0, 6.65, 5.8 or 4.55 kbps, respectively.

Table 1: Bit allocation of the AMR coding algorithm for 20 ms frame

CODING RATE	11.0KBPS	8.0KBPS	6.65KBPS	5.8KBPS	4.55KBPS
Frame size	20ms				
Look ahead	5ms				
LPC order	10 <sup>th</sup> order				
Predictor for LSF	1 predictor, 0 bit/frame				
Quantization	24 bit/frame				
LSF Quantization	2 bit/frame	2 bit/frame	2 bit/frame	0	2 predictors 1 bit/frame
LPC interpolation	0 bit	0	1 bit/frame	0	0
Coding mode bit	0 bit	0 bit	0 bit	0 bit	0 bit
Pitch mode	LTP	LTP	LTP	PP	PP
Subframe size	5ms				
Pitch Lag	30 bit/frame (6956)	8385	8585	0008	0008
Fixed excitation	31 bit/subframe	20	13	18	14 bit/subframe
Gain quantization	9 bits (scalar)		7 bit/subframe		10 bit/subframe
Total	220 bit/frame	160	133	133	91

With reference to Fig. 5, the decoder processing circuitry, pursuant to software control, reconstructs the speech signal using the transmitted modeling indices extracted from the received bit stream by the demultiplexor 511. The decoder processing circuitry decodes the indices to obtain the coder parameters at each transmission frame. These parameters are the LSF vectors, the fractional pitch lags, the innovative code vectors, and the two gains.

The LSF vectors are converted to the LP filter coefficients and interpolated to obtain LP filters at each subframe. At each subframe, the decoder processing circuitry constructs the excitation signal by: 1) identifying the adaptive and innovative code vectors from the codebooks 515 and 519; 2) scaling the contributions by their respective gains at the block 521; 3) summing the scaled contributions; and 3) modifying and applying adaptive tilt compensation at the blocks 527 and 529. The speech signal is also reconstructed on a subframe basis by filtering the excitation through the LP synthesis at the block 531. Finally, the speech signal is passed through an adaptive post filter at the block 535 to generate the reproduced speech signal 539.

The AMR encoder will produce the speech modeling information in a unique sequence and format, and the AMR decoder receives the same information in the same way. The different parameters of the encoded speech and their individual bits have unequal importance with respect



to subjective quality. Before being submitted to the channel encoding function the bits are rearranged in the sequence of importance.

Two pre-processing functions are applied prior to the encoding process: high-pass filtering and signal down-scaling. Down-scaling consists of dividing the input by a factor of 2 to reduce the possibility of overflows in the fixed point implementation. The high-pass filtering at the block 215 (Fig. 2) serves as a precaution against undesired low frequency components. A filter with cut off frequency of 80 Hz is used, and it is given by:

$$H_u(z) = \frac{0.92727435 - 1.8544941z^{-1} + 0.92727435z^{-2}}{1 - 1.9059465z^{-1} + 0.9114024z^{-2}}$$

Down scaling and high-pass filtering are combined by dividing the coefficients of the numerator of  $H_u(z)$  by 2.

Short-term prediction, or linear prediction (LP) analysis is performed twice per speech frame using the autocorrelation approach with 30 ms windows. Specifically, two LP analyses are performed twice per frame using two different windows. In the first LP analysis (LP\_analysis\_1), a hybrid window is used which has its weight concentrated at the fourth subframe. The hybrid window consists of two parts. The first part is half a Hamming window, and the second part is a quarter of a cosine cycle. The window is given by:

$$w_1(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{\pi n}{L}\right) & n = 0 \text{ to } 214, L = 215 \\ \cos\left(\frac{0.49(n-L)\pi}{25}\right) & n = 215 \text{ to } 239 \end{cases}$$

In the second LP analysis (LP\_analysis\_2), a symmetric Hamming window is used.

$$w_2(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{\pi n}{L}\right) & n = 0 \text{ to } 119, L = 120 \\ 0.54 + 0.46 \cos\left(\frac{(n-L)\pi}{120}\right) & n = 120 \text{ to } 239 \end{cases}$$



In either LP analysis, the autocorrelations of the windowed speech  $s'(n)$ ,  $n = 0, 239$  are computed by:

$$r(k) = \sum_{n=k}^{239} s'(n)s'(n-k), k = 0, 10.$$

A 60 Hz bandwidth expansion is used by lag windowing, the autocorrelations using the window:

$$w_{lag}(i) = \exp\left[-\frac{1}{2} \left(\frac{2\pi 60i}{8000}\right)^2\right], i = 1, 10.$$

Moreover,  $r(0)$  is multiplied by a white noise correction factor 1.0001 which is equivalent to adding a noise floor at -40 dB.

The modified autocorrelations  $r'(0) = 1.0001r(0)$ , and  $r'(k) = r(k)w_{lag}(k)$ ,  $k = 1, 10$  are used to obtain the reflection coefficients  $k_i$  and LP filter coefficients  $a_i$ ,  $i = 1, 10$  using the Levinson-Durbin algorithm. Furthermore, the LP filter coefficients  $a_i$  are used to obtain the Line Spectral Frequencies (LSFs).

The interpolated unquantized LP parameters are obtained by interpolating the LSF coefficients obtained from the LP analysis\_1 and those from LP\_analysis\_2 as:

$$\begin{aligned} q_1(n) &= 0.5q_4(n-1) + 0.5q_2(n) \\ q_3(n) &= 0.5q_2(n) + 0.5q_4(n) \end{aligned}$$

where  $q_1(n)$  is the interpolated LSF for subframe 1,  $q_2(n)$  is the LSF of subframe 2 obtained from LP\_analysis\_2 of current frame,  $q_3(n)$  is the interpolated LSF for subframe 3,  $q_4(n-1)$  is the LSF (cosine domain) from LP\_analysis\_1 of previous frame, and  $q_4(n)$  is the LSF for subframe 4 obtained from LP\_analysis\_1 of current frame. The interpolation is carried out in the cosine domain.

A VAD (Voice Activity Detection) algorithm is used to classify input speech frames into either active voice or inactive voice frame (background noise or silence) at a block 235 (Fig. 2).

The input speech  $s(n)$  is used to obtain a weighted speech signal  $s_w(n)$  by passing  $s(n)$  through a filter:

$$W(z) = \frac{A\left(\frac{z}{\gamma_1}\right)}{A\left(\frac{z}{\gamma_2}\right)}.$$

That is, in a subframe of size  $L\_SF$ , the weighted speech is given by:

$$s_w(n) = s(n) + \sum_{i=1}^{10} a_i \gamma_1^i s(n-i) - \sum_{i=1}^{10} a_i \gamma_2^i s_w(n-i), n = 0, L\_SF-1.$$

A voiced/unvoiced classification and mode decision within the block 279 using the input

speech  $s(n)$  and the residual  $r_w(n)$  is derived where:

$$r_w(n) = s(n) + \sum_{i=1}^{10} a_i \gamma_1^i s(n-i), n = 0, L\_SF-1.$$

The classification is based on four measures: 1) speech sharpness  $P1\_SHP$ ; 2) normalized one delay correlation  $P2\_R1$ ; 3) normalized zero-crossing rate  $P3\_ZC$ ; and 4) normalized LP residual energy  $P4\_RE$ .

The speech sharpness is given by:

$$P1\_SHP = \frac{\sum_{n=0}^L abs(r_w(n))}{MaxL}.$$

where  $Max$  is the maximum of  $abs(r_w(n))$  over the specified interval of length  $L$ . The normalized one delay correlation and normalized zero-crossing rate are given by:

$$P2\_R1 = \frac{\sum_{n=0}^{L-1} s(n)s(n+1)}{\sqrt{\sum_{n=0}^{L-1} s(n)s(n)} \sqrt{\sum_{n=0}^{L-1} s(n+1)s(n+1)}}$$

$$P3\_ZC = \frac{1}{2L} \sum_{i=0}^{L-1} [sgn[s(i)] - sgn[s(i-1)]]^2,$$

where  $sgn$  is the sign function whose output is either 1 or -1 depending that the input sample is positive or negative. Finally, the normalized LP residual energy is given by:

$$P4\_RE = 1 - \sqrt{lpc\_gain}$$

where  $lpc\_gain = \prod_{i=1}^{10} (1 - k_i^2)$ , where  $k_i$  are the reflection coefficients obtained from LP analysis\_1.

The voiced/unvoiced decision is derived if the following conditions are met:

if  $P2\_R1 < 0.6$  and  $P1\_SHP > 0.2$  set mode = 2,  
 if  $P3\_ZC > 0.4$  and  $P1\_SHP > 0.18$  set mode = 2,  
 if  $P4\_RE < 0.4$  and  $P1\_SHP > 0.2$  set mode = 2,  
 if  $(P2\_R1 < -1.2 + 3.2P1\_SHP)$  set VUV = -3  
 if  $(P4\_RE < -0.21 + 1.4286P1\_SHP)$  set VUV = -3  
 if  $(P3\_ZC > 0.8 - 0.6P1\_SHP)$  set VUV = -3  
 if  $(P4\_RE < 0.1)$  set VUV = -3

Open loop pitch analysis is performed once or twice (each 10 ms) per frame depending on the coding rate in order to find estimates of the pitch lag at the block 241 (Fig. 2). It is based

on the weighted speech signal  $s_w(n + n_m)$ ,  $n = 0, 1, \dots, 79$ , in which  $n_m$  defines the location of this signal on the first half frame or the last half frame. In the first step, four maxima of the correlation:

$$C_i = \sum_{n=0}^{79} s_w(n_m + n) s_w(n_m + n - k)$$

are found in the four ranges 17, ..., 33, 34, ..., 67, 68, ..., 135, 136, ..., 145, respectively. The retained maxima  $C_{k_i}$ ,  $i = 1, 2, 3, 4$ , are normalized by dividing by:

$$\sqrt{\sum_{n=0}^{79} s_w^2(n_m + n - k)}, \quad i = 1, \dots, 4, \text{ respectively.}$$

The normalized maxima and corresponding delays are denoted by  $(R_{k_i}, k_i)$ ,  $i = 1, 2, 3, 4$ .

In the second step, a delay,  $k_i$ , among the four candidates, is selected by maximizing the four normalized correlations. In the third step,  $k_i$  is probably corrected to  $k_i$  ( $i < 4$ ) by favoring the lower ranges. That is,  $k_i$  ( $i < 4$ ) is selected if  $k_i$  is within  $[k_i/m-4, k_i/m+4]$ ,  $m = 2, 3, 4, 5$ , and if  $k_i > k_j$ ,  $0.95^{i-j} D$ ,  $i < 4$ , where  $D$  is 1.0, 0.85, or 0.65, depending on whether the previous frame is unvoiced, the previous frame is voiced and  $k_i$  is in the neighborhood (specified by  $\pm 8$ ) of the previous pitch lag, or the previous two frames are voiced and  $k_i$  is in the neighborhood of the previous two pitch lags. The final selected pitch lag is denoted by  $T_{op}$ .

A decision is made every frame to either operate the LTP (long-term prediction) as the traditional CELP approach (LTP\_mode=1), or as a modified time warping approach

(LTP\_mode=0) herein referred to as PP (pitch preprocessing). For 4.55 and 5.8 kbps encoding bit rates, LTP\_mode is set to 0 at all times. For 8.0 and 11.0 kbps, LTP\_mode is set to 1 all of the time. Whereas, for a 6.65 kbps encoding bit rate, the encoder decides whether to operate in the LTP or PP mode. During the PP mode, only one pitch lag is transmitted per coding frame.

For 6.65 kbps, the decision algorithm is as follows. First, at the block 241, a prediction of the pitch lag  $pit$  for the current frame is determined as follows:

if (LTP\_MODE\_m == 1)  
 $pit = lagl + 2.4 * (lag\_f[3] - lagl);$   
 else  
 $pit = lag\_f[1] + 2.75 * (lag\_f[3] - lag\_f[1]);$

where LTP\_mode\_m is previous frame LTP\_mode,  $lag\_f[1]$ ,  $lag\_f[3]$  are the past closed loop pitch lags for second and fourth subframes respectively,  $lagl$  is the current frame open-loop pitch lag at the second half of the frame, and  $lagl$  is the previous frame open-loop pitch lag at the first half of the frame.

Second, a normalized spectrum difference between the Line Spectrum Frequencies (LSF) of current and previous frame is computed as:

$$e\_lsf = \frac{1}{10} \sum_{i=0}^9 abs(LSF(i) - LSF\_m(i)),$$

$$\text{if } (abs(pit - lagl) < TH \text{ and } abs(lag\_f[3] - lagl) < lagl * 0.2)$$

$$\text{if } (Rp > 0.5 \text{ \& \& } pgain\_past > 0.7 \text{ and } e\_lsf < 0.5/30) \text{ LTP\_mode} = 0;$$

$$\text{else LTP\_mode} = 1;$$

where  $Rp$  is current frame normalized pitch correlation,  $pgain\_past$  is the quantized pitch gain from the fourth subframe of the past frame,  $TH = MIN(lagl * 0.1, 5)$ , and  $TH = MAX(2.0, TH)$ .

The estimation of the precise pitch lag at the end of the frame is based on the normalized correlation:

$$R_k = \frac{\sum_{n=0}^L s_w(n + n_l) s_w(n + n_l - k)}{\sqrt{\sum_{n=0}^L s_w^2(n + n_l - k)}}$$

where  $s_n(n+n_l)$ ,  $n = 0, 1, \dots, L-1$ , represents the last segment of the weighted speech signal including the look-ahead (the look-ahead length is 25 samples), and the size  $L$  is defined according to the open-loop pitch lag  $T_{op}$  with the corresponding normalized correlation  $C_{T_{op}}$ :

if ( $C_{T_{op}} > 0.6$ )  
 $L = \max(50, T_{op})$   
 $L = \min(80, L)$   
 else  
 $L = 80$

In the first step, one integer lag  $k$  is selected maximizing the  $R_k$  in the range

$k \in [T_{op} - 10, T_{op} + 10]$  bounded by [17, 145]. Then, the precise pitch lag  $P_m$  and the

corresponding index  $I_m$  for the current frame is searched around the integer lag,  $[k-1, k+1]$ , by

up-sampling  $R_k$ .

The possible candidates of the precise pitch lag are obtained from the table named as

*PitLagTab8b[i]*,  $i=0, 1, \dots, 127$ . In the last step, the precise pitch lag  $P_m = \text{PitLagTab8b}[I_m]$  is

possibly modified by checking the accumulated delay  $\tau_{acc}$  due to the modification of the speech

signal:

if ( $\tau_{acc} > 5$ )  $I_m \leftarrow \min(I_m + 1, 127)$ , and  
 if ( $\tau_{acc} < -5$ )  $I_m \leftarrow \max(I_m - 1, 0)$ .

The precise pitch lag could be modified again:

if ( $\tau_{acc} > 10$ )  $I_m \leftarrow \min(I_m + 1, 127)$ , and  
 if ( $\tau_{acc} < -10$ )  $I_m \leftarrow \max(I_m - 1, 0)$ .

The obtained index  $I_m$  will be sent to the decoder:

The pitch lag contour,  $\tau_c(n)$ , is defined using both the current lag  $P_m$  and the previous

lag  $P_{m-1}$ :

-29-

if ( $|P_m - P_{m-1}| < 0.2 \min(P_m, P_{m-1})$ )  
 $\tau_c(n) = P_{m-1} + n(P_m - P_{m-1})/L_f$ ,  $n = 0, 1, \dots, L_f - 1$   
 $\tau_c(n) = P_m$ ,  $n = L_f, \dots, 170$   
 else  
 $\tau_c(n) = P_{m-1}$ ,  $n = 0, 1, \dots, 39$ ;  
 $\tau_c(n) = P_m$ ,  $n = 40, \dots, 170$

where  $L_f = 160$  is the frame size.

One frame is divided into 3 subframes for the long-term preprocessing. For the first two subframes, the subframe size,  $L_n$ , is 53, and the subframe size for searching,  $L_{nr}$ , is 70. For the last subframe,  $L_n$  is 54 and  $L_{nr}$  is:

$$L_{nr} = \min(70, L_n + L_{wd} - 10 - \tau_{acc}),$$

where  $L_{wd}=25$  is the look-ahead and the maximum of the accumulated delay  $\tau_{acc}$  is limited to 14.

The target for the modification process of the weighted speech temporally memorized in

( $\hat{s}_u(m0+n)$ ,  $n = 0, 1, \dots, L_{nr} - 1$ ) is calculated by warping the past modified weighted speech

buffer,  $\hat{s}_u(m0+n)$ ,  $n < 0$ , with the pitch lag contour,  $\tau_c(n+m \cdot L_n)$ ,  $m = 0, 1, 2$ ,

$$\hat{s}_u(m0+n) = \sum_{i=-f_i}^{L_n} \hat{s}_u(m0+n-T_c(n)+i) I_i(i, T_c(n)), \quad n = 0, 1, \dots, L_{nr} - 1,$$

where  $T_c(n)$  and  $T_c(n)$  are calculated by:

$$T_c(n) = \text{round}(\tau_c(n+m \cdot L_n)),$$

$$\bar{T}_c(n) = \tau_c(n) - T_c(n),$$

$m$  is subframe number,  $I_i(i, T_c(n))$  is a set of interpolation coefficients, and  $f_i$  is 10. Then, the

target for matching,  $\hat{s}_i(n)$ ,  $n = 0, 1, \dots, L_{nr} - 1$ , is calculated by weighting

$\hat{s}_u(m0+n)$ ,  $n = 0, 1, \dots, L_{nr} - 1$ , in the time domain:

$$\hat{s}_i(n) = n \cdot \hat{s}_u(m0+n) / L_n, \quad n = 0, 1, \dots, L_n - 1,$$

$$\hat{s}_i(n) = \hat{s}_u(m0+n), \quad n = L_n, \dots, L_{nr} - 1$$

-30-

The local integer shifting range  $(SR0, SRI)$  for searching for the best local delay is computed as the following:

if speech is unvoiced

$$SR0 = -1,$$

$$SRI = 1,$$

else

$$SR0 = \text{round}[-4 \min(1.0, \max(0.0, 1-0.4(P_{sh}-0.2)))]$$

$$SRI = \text{round}[4 \min(1.0, \max(0.0, 1-0.4(P_{sh}-0.2)))]$$

where  $P_{sh} = \max(P_{sh1}, P_{sh2})$ ,  $P_{sh1}$  is the average to peak ratio (i.e., sharpness) from the target signal:

$$P_{sh1} = \frac{\sum_{n=0}^{L_r-1} |\hat{s}_w(m0+n)|}{L_r \max(|\hat{s}_w(m0+n)|, n=0,1,\dots,L_r-1)}$$

and  $P_{sh2}$  is the sharpness from the weighted speech signal:

$$P_{sh2} = \frac{\sum_{n=0}^{L_r-L_r/2-1} |\hat{s}_w(n+n0+L_r/2)|}{(L_r-L_r/2) \max(|\hat{s}_w(n+n0+L_r/2)|, n=0,1,\dots,L_r/2-1)}$$

where  $n0 = \text{trunc}(m0 + \tau_{acc} + 0.5)$  (here,  $m$  is subframe number and  $\tau_{acc}$  is the previous accumulated delay).

In order to find the best local delay,  $\tau_{opt}$ , at the end of the current processing subframe, a normalized correlation vector between the original weighted speech signal and the modified matching target is defined as:

$$R_l(k) = \frac{\sum_{n=0}^{L_r-1} s_w(n0+n+k) \hat{s}_l(n)}{\sqrt{\sum_{n=0}^{L_r-1} s_w^2(n0+n+k) \sum_{n=0}^{L_r-1} \hat{s}_l^2(n)}}$$

A best local delay in the integer domain,  $k_{opt}$ , is selected by maximizing  $R_l(k)$  in the range of  $k \in [SR0, SRI]$ , which is corresponding to the real delay:

$$k_r = k_{opt} + n0 - m0 - \tau_{acc}$$

If  $R_l(k_{opt}) < 0.5$ ,  $k_r$  is set to zero.

In order to get a more precise local delay in the range  $(k_r - 0.75 + 0.1j, j=0,1,\dots,15)$  around  $k_r$ ,  $R_l(k)$  is interpolated to obtain the fractional correlation vector,  $R_f(j)$ , by:

$$R_f(j) = \sum_{i=0}^8 R_l(k_{opt} + L_j + i) I_f(i,j), \quad j = 0,1,\dots,15,$$

where  $\{I_f(i,j)\}$  is a set of interpolation coefficients. The optimal fractional delay index,  $j_{opt}$ , is selected by maximizing  $R_f(j)$ . Finally, the best local delay,  $\tau_{opt}$ , at the end of the current processing subframe, is given by,

$$\tau_{opt} = k_r - 0.75 + 0.1j_{opt}$$

The local delay is then adjusted by:

$$\tau_{opt} = \begin{cases} 0, & \text{if } \tau_{acc} + \tau_{opt} > 14 \\ \tau_{opt}, & \text{otherwise} \end{cases}$$

The modified weighted speech of the current subframe, memorized in

$\{\hat{s}_w(m0+n), n=0,1,\dots,L_r-1\}$  to update the buffer and produce the second target signal 253 for searching the fixed codebook 261, is generated by warping the original weighted speech  $\{s_w(n)\}$  from the original time region,

$$[m0 + \tau_{acc}, m0 + \tau_{acc} + L_r + \tau_{opt}],$$

to the modified time region,

$$[m0, m0 + L_r]:$$

$$\hat{x}_n(m0+n) = \sum_{i=m-L_f+1}^L s_n(m0+n+T_w(n)+i) I_f(i, T_w(n)), \quad n = 0, 1, \dots, L_f - 1,$$

where  $T_w(n)$  and  $T_{nw}(n)$  are calculated by:

$$T_w(n) = \text{trunc}(\tau_{acc} + n \cdot \tau_{op} / L_f),$$

$$T_{nw}(n) = \tau_{acc} + n \cdot \tau_{op} / L_f - T_w(n),$$

$\{I_f(i, T_w(n))\}$  is a set of interpolation coefficients.

After having completed the modification of the weighted speech for the current subframe,

the modified target weighted speech buffer is updated as follows:

$$\hat{x}_n(n) \Leftarrow \hat{x}_n(n + L_f), \quad n = 0, 1, \dots, n_m - 1.$$

The accumulated delay at the end of the current subframe is renewed by:

$$\tau_{acc} \Leftarrow \tau_{acc} + \tau_{op}.$$

Prior to quantization the LSFs are smoothed in order to improve the perceptual quality.

In principle, no smoothing is applied during speech and segments with rapid variations in the spectral envelope. During non-speech with slow variations in the spectral envelope, smoothing is applied to reduce unwanted spectral variations. Unwanted spectral variations could typically occur due to the estimation of the LPC parameters and LSF quantization. As an example, in stationary noise-like signals with constant spectral envelope introducing even very small variations in the spectral envelope is picked up easily by the human ear and perceived as an annoying modulation.

The smoothing of the LSFs is done as a running mean according to:

$$lsf_i(n) = \beta(n) \cdot lsf_i(n-1) + (1 - \beta(n)) \cdot lsf\_est_i(n), \quad i = 1, \dots, 10$$

where  $lsf\_est_i(n)$  is the  $i^{\text{th}}$  estimated LSF of frame  $n$ , and  $lsf_i(n)$  is the  $i^{\text{th}}$  LSF for quantization of frame  $n$ . The parameter  $\beta(n)$  controls the amount of smoothing, e.g. if  $\beta(n)$  is zero no smoothing is applied.

$\beta(n)$  is calculated from the VAD information (generated at the block 235) and two estimates of the evolution of the spectral envelope. The two estimates of the evolution are defined as:

$$\Delta SP = \sum_{n=1}^{10} (lsf\_est_i(n) - lsf\_est_i(n-1))^2$$

$$\Delta SP_{\text{inv}} = \sum_{n=1}^{10} (lsf\_est_i(n) - ma\_lsf_i(n-1))^2$$

$$ma\_lsf_i(n) = \beta(n) \cdot ma\_lsf_i(n-1) + (1 - \beta(n)) \cdot lsf\_est_i(n), \quad i = 1, \dots, 10$$

The parameter  $\beta(n)$  is controlled by the following logic:

Step 1:

```

if (Vad = 1 | PassVad = 1 |  $k_1 > 0.5$ )
     $N_{mode\_sm}(n-1) = 0$ 
 $\beta(n) = 0.0$ 
elseif ( $N_{mode\_sm}(n-1) > 0$  & ( $\Delta SP > 0.0015$  |  $\Delta SP_{int} > 0.0024$ ))
     $N_{mode\_sm}(n-1) = 0$ 
 $\beta(n) = 0.0$ 
elseif ( $N_{mode\_sm}(n-1) > 1$  &  $\Delta SP > 0.0025$ )
     $N_{mode\_sm}(n-1) = 1$ 
endif

```

Step 2:

```

if (Vad = 0 & PassVad = 0)
     $N_{mode\_sm}(n) = N_{mode\_sm}(n-1) + 1$ 
    if ( $N_{mode\_sm}(n) > 5$ )
         $N_{mode\_sm}(n) = 5$ 
    endif
     $\beta(n) = \frac{0.9}{16} \cdot (N_{mode\_sm}(n) - 1)^2$ 
else
     $N_{mode\_sm}(n) = N_{mode\_sm}(n-1)$ 
endif

```

where  $k_1$  is the first reflection coefficient.

In step 1, the encoder processing circuitry checks the VAD and the evolution of the spectral envelope, and performs a full or partial reset of the smoothing if required. In step 2, the encoder processing circuitry updates the counter,  $N_{mode\_sm}(n)$ , and calculates the smoothing parameter,  $\beta(n)$ . The parameter  $\beta(n)$  varies between 0.0 and 0.9, being 0.0 for speech, music,

tonal-like signals, and non-stationary background noise and ramping up towards 0.9 when stationary background noise occurs.

The LSFs are quantized once per 20 ms frame using a predictive multi-stage vector quantization. A minimal spacing of 50 Hz is ensured between each two neighboring LSFs before quantization. A set of weights is calculated from the LSFs, given by  $w_i = K|P(f_i)|^{0.4}$  where  $f_i$  is the  $i^{th}$  LSF value and  $P(f_i)$  is the LPC power spectrum at  $f_i$  ( $K$  is an irrelevant multiplicative constant). The reciprocal of the power spectrum is obtained by (up to a multiplicative constant):

$$P(f_i)^{-1} = \begin{cases} (1 - \cos(2\pi f_i)) \prod_{\text{odd}} [\cos(2\pi f_i') - \cos(2\pi f_i')]^2 & \text{even } i \\ (1 + \cos(2\pi f_i)) \prod_{\text{even}} [\cos(2\pi f_i') - \cos(2\pi f_i')]^2 & \text{odd } i \end{cases}$$

and the power of  $-0.4$  is then calculated using a lookup table and cubic-spline interpolation between table entries.

A vector of mean values is subtracted from the LSFs, and a vector of prediction error vector  $fe$  is calculated from the mean removed LSFs vector, using a full-matrix AR(2) predictor. A single predictor is used for the rates 5.8, 6.65, 8.0, and 11.0 kbps coders, and two sets of prediction coefficients are tested as possible predictors for the 4.55 kbps coder.

The vector of prediction error is quantized using a multi-stage VQ, with multi-surviving candidates from each stage to the next stage. The two possible sets of prediction error vectors generated for the 4.55 kbps coder are considered as surviving candidates for the first stage.

The first 4 stages have 64 entries each, and the fifth and last table have 16 entries. The first 3 stages are used for the 4.55 kbps coder, the first 4 stages are used for the 5.8, 6.65 and 8.0 kbps coders, and all 5 stages are used for the 11.0 kbps coder. The following table summarizes the number of bits used for the quantization of the LSFs for each rate.

	prediction	1 <sup>st</sup> stage	2 <sup>nd</sup> stage	3 <sup>rd</sup> stage	4 <sup>th</sup> stage	5 <sup>th</sup> stage	total
4.55 kbps	1	6	6	6			19
5.8 kbps	0	6	6	6	6		24
6.65 kbps	0	6	6	6	6		24
8.0 kbps	0	6	6	6	6		24
11.0 kbps	0	6	6	6	6	4	28

The number of surviving candidates for each stage is summarized in the following table.

	prediction candidates into the 1 <sup>st</sup> stage	Surviving candidates from the 1 <sup>st</sup> stage	surviving candidates from the 2 <sup>nd</sup> stage	surviving candidates from the 3 <sup>rd</sup> stage	surviving candidates from the 4 <sup>th</sup> stage
4.55 kbps	2	10	6	4	
5.8 kbps	1	8	6	4	
6.65 kbps	1	8	8	4	
8.0 kbps	1	8	8	4	
11.0 kbps	1	8	6	4	4

The quantization in each stage is done by minimizing the weighted distortion measure

given by:

$$e_k = \sum_{i=0}^2 w_i (f_i - c_i^k)^2$$

The code vector with index  $k_{\min}$  which minimizes  $e_k$  such that  $e_{k_{\min}} < e_k$  for all  $k$ , is chosen to represent the prediction/quantization error ( $f_e$  represents in this equation both the initial prediction error to the first stage and the successive quantization error from each stage to the next one).

The final choice of vectors from all of the surviving candidates (and for the 4.55 kbps coder - also the predictor) is done at the end, after the last stage is searched, by choosing a

combined set of vectors (and predictor) which minimizes the total error. The contribution from all of the stages is summed to form the quantized prediction error vector, and the quantized prediction error is added to the prediction states and the mean LSFs value to generate the quantized LSFs vector.

For the 4.55 kbps coder, the number of order flips of the LSFs as the result of the

quantization if counted, and if the number of flips is more than 1, the LSFs vector is replaced with 0.9 · (LSFs of previous frame) + 0.1 · (mean LSFs value). For all the rates, the quantized

LSFs are ordered and spaced with a minimal spacing of 50 Hz.

The interpolation of the quantized LSF is performed in the cosine domain in two ways depending on the LTP\_mode. If the LTP\_mode is 0, a linear interpolation between the quantized LSF set of the current frame and the quantized LSF set of the previous frame is performed to get the LSF set for the first, second and third subframes as:

$$\bar{q}_1(n) = 0.75\bar{q}_4(n-1) + 0.25\bar{q}_4(n)$$

$$\bar{q}_2(n) = 0.5\bar{q}_4(n-1) + 0.5\bar{q}_4(n)$$

$$\bar{q}_3(n) = 0.25\bar{q}_4(n-1) + 0.75\bar{q}_4(n)$$

where  $\bar{q}_1(n-1)$  and  $\bar{q}_4(n)$  are the cosines of the quantized LSF sets of the previous and current frames, respectively, and  $\bar{q}_1(n)$ ,  $\bar{q}_2(n)$  and  $\bar{q}_3(n)$  are the interpolated LSF sets in cosine domain for the first, second and third subframes respectively.

If the LTP\_mode is 1, a search of the best interpolation path is performed in order to get the interpolated LSF sets. The search is based on a weighted mean absolute difference between a reference LSF set  $\bar{r}_1(n)$  and the LSF set obtained from LP analysis,  $2 \cdot \bar{l}(n)$ . The weights  $\bar{w}$  are computed as follows:



$$w(0) = (1 - l(0))(1 - l(1) + l(0))$$

$$w(9) = (1 - l(9))(1 - l(9) + l(8))$$

for  $t = 1$  to 9

$$w(i) = (1 - l(i))(1 - \text{Min}(l(i+1) - l(i), l(i) - l(i-1)))$$

where  $\text{Min}(a, b)$  returns the smallest of  $a$  and  $b$ .

There are four different interpolation paths. For each path, a reference LSF set  $r\bar{q}(n)$  in

cosine domain is obtained as follows:

$$r\bar{q}_k(n) = \alpha(k)\bar{q}_k(n) + (1 - \alpha(k))\bar{q}_k(n-1), k = 1 \text{ to } 4$$

$\bar{\alpha} = \{0.4, 0.5, 0.6, 0.7\}$  for each path respectively. Then the following distance measure is

computed for each path as:

$$D = |r\bar{q}(n) - \bar{q}(n)|^T \bar{w}$$

The path leading to the minimum distance  $D$  is chosen and the corresponding reference LSF set

$r\bar{q}(n)$  is obtained as:

$$r\bar{q}(n) = \alpha_{op}\bar{q}_k(n) + (1 - \alpha_{op})\bar{q}_k(n-1)$$

The interpolated LSF sets in the cosine domain are then given by:

$$\bar{q}_1(n) = 0.5\bar{q}_k(n-1) + 0.5r\bar{q}(n)$$

$$\bar{q}_2(n) = r\bar{q}(n)$$

$$\bar{q}_3(n) = 0.5r\bar{q}(n) + 0.5\bar{q}_k(n)$$

The impulse response,  $h(n)$ , of the weighted synthesis filter

$$H(z)W(z) = A(z/\gamma_1)/(\bar{A}(z)A(z/\gamma_2))$$
 is computed each subframe. This impulse response is

needed for the search of adaptive and fixed codebooks 257 and 261. The impulse response

$h(n)$  is computed by filtering the vector of coefficients of the filter  $A(z/\gamma_1)$  extended by zeros through the two filters  $1/\bar{A}(z)$  and  $1/A(z/\gamma_2)$ .

The target signal for the search of the adaptive codebook 257 is usually computed by subtracting the zero input response of the weighted synthesis filter  $H(z)W(z)$  from the weighted speech signal  $s_-(n)$ . This operation is performed on a frame basis. An equivalent procedure for computing the target signal is the filtering of the LP residual signal  $r(n)$  through the combination of the synthesis filter  $1/\bar{A}(z)$  and the weighting filter  $W(z)$ .

After determining the excitation for the subframe, the initial states of these filters are updated by filtering the difference between the LP residual and the excitation. The LP residual is given by:

$$r(n) = s(n) + \sum_{i=1}^{10} \bar{a}_i s(n-i), n = 0, L\_SF - 1$$

The residual signal  $r(n)$  which is needed for finding the target vector is also used in the adaptive codebook search to extend the past excitation buffer. This simplifies the adaptive codebook search procedure for delays less than the subframe size of 40 samples.

In the present embodiment, there are two ways to produce an LTP contribution. One uses pitch preprocessing (PP) when the PP-mode is selected, and another is computed like the traditional LTP when the LTP-mode is chosen. With the PP-mode, there is no need to do the adaptive codebook search, and LTP excitation is directly computed according to past synthesized excitation because the interpolated pitch contour is set for each frame. When the AMR coder operates with LTP-mode, the pitch lag is constant within one subframe, and searched and coded on a subframe basis.

Suppose the past synthesized excitation is memorized in  $/\text{ext}(\text{MAX\_LAG}+n)$ ,  $n < 0$ , which is also called adaptive codebook. The LTP excitation codevector, temporally memorized in  $/\text{ext}(\text{MAX\_LAG}+n)$ ,  $0 \leq n < L\_SF$ , is calculated by interpolating the past excitation (adaptive

codebook) with the pitch lag contour,  $\tau_c(n+m \cdot L\_SF)$ ,  $m=0,1,2,3$ . The interpolation is performed using an FIR filter (Hamming windowed sinc functions):

$$ext(MAX\_LAG+n) = \sum_{m=-f_i}^f ext(MAX\_LAG+n - \tau_c(n) + i \cdot l_i(i \cdot T_{ic}(n))), n=0,1,\dots,L\_SF-1;$$

where  $T_c(n)$  and  $T_{ic}(n)$  are calculated by

$$T_c(n) = \text{trunc}(\tau_c(n+m \cdot L\_SF)),$$

$$T_{ic}(n) = \tau_c(n) - T_c(n),$$

$m$  is subframe number,  $(l_i(i \cdot T_{ic}(n)))$  is a set of interpolation coefficients,  $f_i$  is 10,  $MAX\_LAG$  is 145+1, and  $L\_SF=40$  is the subframe size. Note that the interpolated values

$(ext(MAX\_LAG+n), 0 \leq n < L\_SF - 17 + 11)$  might be used again to do the interpolation when the pitch lag is small. Once the interpolation is finished, the adaptive codevector  $V_a = (v_d(n), n=0$  to 39) is obtained by copying the interpolated values:

$$v_d(n) = ext(MAX\_LAG+n), 0 \leq n < L\_SF$$

Adaptive codebook searching is performed on a subframe basis. It consists of performing closed-loop pitch lag search, and then computing the adaptive code vector by interpolating the past excitation at the selected fractional pitch lag. The LTP parameters (or the adaptive codebook parameters) are the pitch lag (or the delay) and gain of the pitch filter. In the search stage, the excitation is extended by the LP residual to simplify the closed-loop search.

For the bit rate of 11.0 kbps, the pitch delay is encoded with 9 bits for the 1<sup>st</sup> and 3<sup>rd</sup> subframes and the relative delay of the other subframes is encoded with 6 bits. A fractional pitch delay is used in the first and third subframes with resolutions:  $1/6$  in the range  $[17, 93 - \frac{4}{6}]$ , and integers only in the range [95, 145]. For the second and fourth subframes, a pitch resolution of

$1/6$  is always used for the rate 11.0 kbps in the range  $[T_1 - \frac{3}{6}, T_1 + \frac{3}{6}]$ , where  $T_1$  is the pitch lag of the previous (1<sup>st</sup> or 3<sup>rd</sup>) subframe.

The close-loop pitch search is performed by minimizing the mean-square weighted error between the original and synthesized speech. This is achieved by maximizing the term:

$$R(k) = \frac{\sum_{n=0}^{39} T_{\mu}(n) y_k(n)}{\sqrt{\sum_{n=0}^{39} y_k(n) y_k(n)}}, \text{ where } T_{\mu}(n) \text{ is the target signal and } y_k(n) \text{ is the past filtered}$$

excitation at delay  $k$  (past excitation convoluted with  $h(n)$ ). The convolution  $y_k(n)$  is

computed for the first delay  $l_{ms}$  in the search range, and for the other delays in the search range

$k = l_{ms} + 1, \dots, l_{ms}$ , it is updated using the recursive relation:

$$y_k(n) = y_{k-1}(n-1) + u(-)h(n),$$

where  $u(n), n = -(143+11)$  to 39 is the excitation buffer.

Note that in the search stage, the samples  $u(n), n = 0$  to 39, are not available and are needed for pitch delays less than 40. To simplify the search, the LP residual is copied to  $u(n)$  to make the relation in the calculations valid for all delays. Once the optimum integer pitch delay is determined, the fractions, as defined above, around that integer are tested. The fractional pitch search is performed by interpolating the normalized correlation and searching for its maximum.

Once the fractional pitch lag is determined, the adaptive codebook vector,  $v(n)$ , is computed by interpolating the past excitation  $u(n)$  at the given phase (fraction). The interpolations are performed using two FIR filters (Hamming windowed sinc functions), one for interpolating the term in the calculations to find the fractional pitch lag and the other for

interpolating the past excitation as previously described. The adaptive codebook gain,  $g_p$ , is temporally given then by:

$$g_p = \frac{\sum_{n=0}^{19} I_p(n) y(n)}{\sum_{n=0}^{19} y(n) y(n)},$$

bounded by  $0 < g_p < 1.2$ , where  $y(n) = v(n) * h(n)$  is the filtered adaptive

codebook vector (zero state response of  $H(z)W(z)$  to  $v(n)$ ). The adaptive codebook gain could be modified again due to joint optimization of the gains, gain normalization and smoothing. The term  $y(n)$  is also referred to herein as  $C_p(n)$ .

With conventional approaches, pitch lag maximizing correlation might result in two or more times the correct one. Thus, with such conventional approaches, the candidate of shorter pitch lag is favored by weighting the correlations of different candidates with constant weighting coefficients. At times this approach does not correct the double or treble pitch lag because the weighting coefficients are not aggressive enough or could result in halving the pitch lag due to the strong weighting coefficients.

In the present embodiment, these weighting coefficients become adaptive by checking if the present candidate is in the neighborhood of the previous pitch lags (when the previous frames are voiced) and if the candidate of shorter lag is in the neighborhood of the value obtained by dividing the longer lag (which maximizes the correlation) with an integer.

In order to improve the perceptual quality, a speech classifier is used to direct the searching procedure of the fixed codebook (as indicated by the blocks 275 and 279) and to-control gain normalization (as indicated in the block 401 of Fig. 4). The speech classifier serves to improve the background noise performance for the lower rate coders, and to get a quick start-

up of the noise level estimation. The speech classifier distinguishes stationary noise-like segments from segments of speech, music, tonal-like signals, non-stationary noise, etc.

The speech classification is performed in two steps. An initial classification (*speech\_mode*) is obtained based on the modified input signal. The final classification (*exc\_mode*) is obtained from the initial classification and the residual signal after the pitch contribution has been removed. The two outputs from the speech classification are the excitation mode, *exc\_mode*, and the parameter  $\beta_{sub}(n)$ , used to control the subframe based smoothing of the gains.

The speech classification is used to direct the encoder according to the characteristics of the input signal and need not be transmitted to the decoder. Thus, the bit allocation, codebooks, and decoding remain the same regardless of the classification. The encoder emphasizes the perceptually important features of the input signal on a subframe basis by adapting the encoding in response to such features. It is important to notice that misclassification will not result in disastrous speech quality degradations. Thus, as opposed to the VAD 235, the speech classifier identified within the block 279 (Fig. 2) is designed to be somewhat more aggressive for optimal perceptual quality.

The initial classifier (*speech\_classifier*) has adaptive thresholds and is performed in six steps:

1. Adapt thresholds:

if (*updates\_noise*  $\geq$  30 & *updates\_speech*  $\geq$  30)

$$SNR\_max = \min \left( \frac{ma\_max\_speech}{ma\_max\_noise}, 32 \right)$$

else

$$SNR\_max = 3.5$$

endif

if (*SNR\_max* < 1.75)

$$dec\_max\_mes = 1.30$$

$$dec\_ma\_cp = 0.70$$

$$update\_max\_mes = 1.10$$

$$update\_ma\_cp\_speech = 0.72$$

elseif (*SNR\_max* < 2.50)

$$dec\_max\_mes = 1.65$$

$$dec\_ma\_cp = 0.73$$

$$update\_max\_mes = 1.30$$

$$update\_ma\_cp\_speech = 0.72$$

else

$$dec\_max\_mes = 1.75$$

$$dec\_ma\_cp = 0.77$$

$$update\_max\_mes = 1.30$$

$$update\_ma\_cp\_speech = 0.77$$

endif

2. Calculate parameters:

Pitch correlation:

$$cp = \frac{\sum_{i=0}^{L_{SF}-1} \tilde{s}(i) \cdot \tilde{s}(i - lag)}{\sqrt{\left( \sum_{i=0}^{L_{SF}-1} \tilde{s}(i) \cdot \tilde{s}(i) \right) \cdot \left( \sum_{i=0}^{L_{SF}-1} \tilde{s}(i - lag) \cdot \tilde{s}(i - lag) \right)}}$$

Running mean of pitch correlation:

$$ma\_cp(n) = 0.9 \cdot ma\_cp(n-1) + 0.1 \cdot cp$$

Maximum of signal amplitude in current pitch cycle:

$$max(n) = \max\{\tilde{s}(i) \mid i = start, \dots, L_{SF} - 1\}$$

where:

$$start = \min\{L_{SF} - lag, 0\}$$

Sum of signal amplitudes in current pitch cycle:

$$mean(n) = \sum_{i=start}^{L_{SF}-1} \tilde{s}(i)$$

Measure of relative maximum:

$$max\_mes = \frac{max(n)}{ma\_max\_noise(n-1)}$$

Maximum to long-term sum:

$$max2sum = \frac{max(n)}{\sum_{k=1}^{14} mean(n-k)}$$

Maximum in groups of 3 subframes for past 15 subframes:

$$max\_group(n, k) = \max\{max(n-3 \cdot (4-k) - j) \mid j = 0, \dots, 2\} \quad k = 0, \dots, 4$$

Group-maximum to minimum of previous 4 group-maxima:

$$endmax2minmax = \frac{max\_group(n, 4)}{\min\{max\_group(n, k) \mid k = 0, \dots, 3\}}$$

Slope of 5 group maxima:

$$slope = 0.1 \cdot \sum_{i=0}^4 (i-2) \cdot max\_group(n, i)$$

## 3. Classify subframe:

```

if (((max_mes < deci_max_mes & ma_cp < deci_ma_cp) | (VAD = 0)) &
    (LTP_MODE = 1 | 5.8kbit/s | 4.55kbit/s))
    speech_mode = 0 /* class1 */
else
    speech_mode = 1 /* class2 */
endif

```

## 4. Check for change in background noise level, i.e. reset required:

Check for decrease in level:

```

if (updates_noise = 31 & max_mes <= 0.3)
    if (consec_low < 15)
        consec_low++
    endif
else
    consec_low = 0
endif

if (consec_low = 15)
    updates_noise = 0
    lev_reset = -1 /* low level reset */
endif

```

Check for increase in level:

```

if ((updates_noise >= 30 | lev_reset = -1) & max_mes > 1.5 & ma_cp < 0.70 & cp < 0.85
    & k1 < -0.4 & endmax2minmax < 50 & max2sum < 35 & slope > -100 & slope < 120)
    if (consec_high < 15)
        consec_high++
    endif
else
    consec_high = 0
endif

if (consec_high = 15 & endmax2minmax < 6 & max2sum < 5)
    updates_noise = 30
    lev_reset = 1 /* high level reset */
endif

```

## 5. Update running mean of maximum of class 1 segments, i.e. stationary noise:

```

if (
    /* 1. condition : regular update */
    (max_mes < update_max_mes & ma_cp < 0.6 & cp < 0.65 & max_mes > 0.3) |
    /* 2. condition : VAD continued update */
    (consec_vad_0 = 8) |
    /* 3. condition : start - up/reset update */
    (updates_noise <= 30 & ma_cp < 0.7 & cp < 0.75 & k1 < -0.4 & endmax2minmax < 5 &
    (lev_reset = -1) | (lev_reset = -1 & max_mes < 2)))
    ma_max_noise(n) = 0.9 * ma_max_noise(n-1) + 0.1 * max(n)
endif

if (updates_noise <= 30)
    updates_noise ++
else
    lev_reset = 0
endif

```

where  $k_1$  is the first reflection coefficient.

## 6. Update running mean of maximum of class 2 segments, i.e. speech, music, tonal-like signals, non-stationary noise, etc, continued from above:

```

elseif (ma_cp > update_ma_cp_speech)
    if (updates_speech <= 80)
         $\alpha_{speech} = 0.95$ 
    else
         $\alpha_{speech} = 0.999$ 
    endif

    ma_max_speech(n) =  $\alpha_{speech} \cdot ma_{max\_speech}(n-1) + (1 - \alpha_{speech}) \cdot max(n)$ 

    if (updates_speech <= 80)
        updates_speech ++
    endif

```

The final classifier (*exc\_preselct*) provides the final class, *exc\_mode*, and the subframe based smoothing parameter,  $\beta_{sub}(n)$ . It has three steps:

1. Calculate parameters:

Maximum amplitude of ideal excitation in current subframe:

$$\max_{res}(n) = \max\{res2(i) | i = 0, \dots, L_{SF} - 1\}$$

Measure of relative maximum:

$$\max\_mes_{res} = \frac{\max_{res}(n)}{\max_{res}(n) - 1}$$

2. Classify subframe and calculate smoothing:

if (*speech\_mode* = 1 |  $\max\_mes_{res} \geq 1.75$ )

*exc\_mode* = 1 / \*class 2 \*/

$\beta_{sub}(n) = 0$

*N\_mode\_sub*(*n*) = -4

else

*exc\_mode* = 0 / \*class 1 \*/

*N\_mode\_sub*(*n*) = *N\_mode\_sub*(*n* - 1) + 1

if (*N\_mode\_sub*(*n*) > 4)

*N\_mode\_sub*(*n*) = 4

endif

if (*N\_mode\_sub*(*n*) > 0)

$$\beta_{sub}(n) = \frac{0.7}{9} \cdot (N\_mode\_sub(n) - 1)^2$$

else

$\beta_{sub}(n) = 0$

endif

endif

3. Update running mean of maximum:

if ( $\max\_mes_{res} \leq 0.5$ )

if (*consec* < 51)

*consec* ++

endif

else

*consec* = 0

endif

if ((*exc\_mode* = 0 & ( $\max\_mes_{res} > 0.5$  | *consec* > 50)) |

(*updates* ≤ 30 & *ma\_cp* < 0.6 & *cp* < 0.65))

*ma\_max*(*n*) = 0.9 · *ma\_max*(*n* - 1) + 0.1 ·  $\max_{res}(n)$

if (*updates* ≤ 30)

*updates* ++

endif

endif

When this process is completed, the final subframe based classification, *exc\_mode*, and the smoothing parameter,  $\beta_{sub}(n)$ , are available.

To enhance the quality of the search of the fixed codebook 261, the target signal,  $T_g(n)$ , is produced by temporally reducing the LTP contribution with a gain factor,  $G_r$ :

$$T_g(n) = T_g(n) \cdot G_r \cdot g_p \cdot Y_g(n), \quad n=0, 1, \dots, 39$$

where  $T_g(n)$  is the original target signal 253,  $Y_g(n)$  is the filtered signal from the adaptive codebook,  $g_p$  is the LTP gain for the selected adaptive codebook vector, and the gain factor is determined according to the normalized LTP gain,  $R_p$ , and the bit rate:

if (*rate* ≤ 0) /\*for 4.45kbps and 5.8kbps\*/

$G_r = 0.7 R_p + 0.3$ ;

if (*rate* = 1) /\*for 6.65kbps \*/

$G_r = 0.6 R_p + 0.4$ ;

if (rate==2) /\* for 8.0kbps \*/  
 $G_r = 0.3 R_p + 0.7;$

if (rate==3) /\* for 11.0kbps \*/  
 $G_r = 0.95;$

if ( $T_{op} > L_{SF}$  &  $g_p > 0.5$  &  $rate \leq 2$ )  
 $G_r \Leftarrow G_r \cdot (0.3 \cdot R_p + 0.7);$  and

where normalized LTP gain,  $R_p$ , is defined as:

$$R_p = \frac{\sum_{n=0}^{39} T_p(n) Y_a(n)}{\sqrt{\sum_{n=0}^{39} T_p(n) T_p(n) \sum_{n=0}^{39} Y_a(n) Y_a(n)}}$$

Another factor considered at the control block 275 in conducting the fixed codebook search and at the block 401 (Fig. 4) during gain normalization is the noise level + ")", which is given by:

$$P_{MSR} = \sqrt{\frac{\max((E_n - 100), 0.0)}{E_i}}$$

where  $E_i$  is the energy of the current input signal including background noise, and  $E_n$  is a running average energy of the background noise.  $E_n$  is updated only when the input signal is detected to be background noise as follows:

if (first background noise frame is true)  
 $E_n = 0.75 E_i;$   
else if (background noise frame is true)  
 $E_n = 0.75 E_{n,m} + 0.25 E_i;$

where  $E_{n,m}$  is the last estimation of the background noise energy.

For each bit rate mode, the fixed codebook 261 (Fig. 2) consists of two or more subcodebooks which are constructed with different structure. For example, in the present embodiment at higher rates, all the subcodebooks only contain pulses. At lower bit rates, one of

the subcodebooks is populated with Gaussian noise. For the lower bit-rates (e.g., 6.65, 5.8, 4.55 kbps), the speech classifier forces the encoder to choose from the Gaussian subcodebook in case of stationary noise-like subframes,  $exc\_mode = 0$ . For  $exc\_mode = 1$  all subcodebooks are searched using adaptive weighting.

For the pulse subcodebooks, a fast searching approach is used to choose a subcodebook and select the code word for the current subframe. The same searching routine is used for all the bit rate modes with different input parameters.

In particular, the long-term enhancement filter,  $F_p(z)$ , is used to filter through the selected pulse excitation. The filter is defined as  $F_p(z) = \frac{1}{(1 - \beta z^{-T})}$ , where  $T$  is the integer part of pitch lag at the center of the current subframe, and  $\beta$  is the pitch gain of previous subframe, bounded by [0.2, 1.0]. Prior to the codebook search, the impulsive response  $h(n)$  includes the filter  $F_p(z)$ .

For the Gaussian subcodebooks, a special structure is used in order to bring down the storage requirement and the computational complexity. Furthermore, no pitch enhancement is applied to the Gaussian subcodebooks.

There are two kinds of pulse subcodebooks in the present AMR coder embodiment. All pulses have the amplitudes of +1 or -1. Each pulse has 0, 1, 2, 3 or 4 bits to code the pulse position. The signs of some pulses are transmitted to the decoder with one bit coding one sign. The signs of other pulses are determined in a way related to the coded signs and their pulse positions.

In the first kind of pulse subcodebook, each pulse has 3 or 4 bits to code the pulse position. The possible locations of individual pulses are defined by two basic non-regular tracks and initial phases:

$$POS(n_p, i) = TRACK(m_p, i) + PHAS(n_p, phase\_mode),$$

where  $i=0, 1, \dots, 7$  or  $15$  (corresponding to 3 or 4 bits to code the position), is the possible position index,  $n_p = 0, \dots, N_p-1$  ( $N_p$  is the total number of pulses), distinguishes different pulses,  $m_p=0$  or  $1$ , defines two tracks, and  $phase\_mode=0$  or  $1$ , specifies two phase modes.

For 3 bits to code the pulse position, the two basic tracks are:

$$TRACK(0, i) = \{0, 4, 8, 12, 16, 24, 30, 36\}, \text{ and} \\ TRACK(1, i) = \{0, 6, 12, 18, 22, 26, 30, 34\}.$$

If the position of each pulse is coded with 4 bits, the basic tracks are:

$$TRACK(0, i) = \{0, 2, 4, 6, 8, 10, 12, 14, 16, 17, 20, 23, 26, 29, 32, 35, 36\}, \text{ and} \\ TRACK(1, i) = \{0, 3, 6, 9, 12, 15, 18, 21, 23, 25, 27, 29, 31, 33, 35, 37\}.$$

The initial phase of each pulse is fixed as:

$$PHAS(n_p, 0) = \text{modulus}(n_p / MAXPHAS) \\ PHAS(n_p, 1) = PHAS(N_p - 1 - n_p, 0)$$

where  $MAXPHAS$  is the maximum phase value.

For any pulse subcodebook, at least the first sign for the first pulse,  $SIGN(n_p)$ ,  $n_p=0$ , is encoded because the gain sign is embedded. Suppose  $N_{sig}$  is the number of pulses with encoded signs; that is,  $SIGN(n_p)$ , for  $n_p < N_{sig}$ ,  $\leq N_p$ , is encoded while  $SIGN(n_p)$ , for  $n_p > N_{sig}$ , is not encoded. Generally, all the signs can be determined in the following way:

$$SIGN(n_p) = -SIGN(n_p-1), \text{ for } n_p > N_{sig}$$

due to that the pulse positions are sequentially searched from  $n_p=0$  to  $n_p=N_p-1$  using an iteration approach. If two pulses are located in the same track while only the sign of the first pulse in the track is encoded, the sign of the second pulse depends on its position relative to the first pulse. If the position of the second pulse is smaller, then it has opposite sign, otherwise it has the same sign as the first pulse.

In the second kind of pulse subcodebook, the innovation vector contains 10 signed pulses. Each pulse has 0, 1, or 2 bits to code the pulse position. One subframe with the size of 40 samples is divided into 10 small segments with the length of 4 samples. 10 pulses are respectively located into 10 segments. Since the position of each pulse is limited into one segment, the possible locations for the pulse numbered with  $n_p$  are,  $(4n_p)$ ,  $(4n_p+2)$ , or  $(4n_p+4n_p+1)$ ,  $(4n_p+2)$ ,  $(4n_p+3)$ , respectively for 0, 1, or 2 bits to code the pulse position. All the signs for all the 10 pulses are encoded.

The fixed codebook 261 is searched by minimizing the mean square error between the weighted input speech and the weighted synthesized speech. The target signal used for the LTP excitation is updated by subtracting the adaptive codebook contribution. That is:

$$x_2(n) = x(n) - \hat{g}_p y(n), \quad n=0, \dots, 39,$$

where  $y(n) = y(n) * h(n)$  is the filtered adaptive codebook vector and  $\hat{g}_p$  is the modified (reduced) LTP gain.

If  $c_k$  is the code vector at index  $k$  from the fixed codebook, then the pulse codebook is searched by maximizing the term:

$$A_k = \frac{(C_k)^2}{E_{Dk}} = \frac{(d^T c_k)^2}{c_k^T \Phi c_k},$$

where  $d = H^T x_1$  is the correlation between the target signal  $x_1(n)$  and the impulse response  $h(n)$ ,  $H$  is a lower triangular Toeplitz convolution matrix with diagonal  $h(0)$  and lower diagonals  $h(1), \dots, h(39)$ , and  $\Phi = H^T H$  is the matrix of correlations of  $h(n)$ . The vector  $d$  (backward filtered target) and the matrix  $\Phi$  are computed prior to the codebook search. The elements of the vector  $d$  are computed by:



$$d(n) = \sum_{i=0}^{39} x_2(i)H(i-n), \quad n=0, \dots, 39,$$

and the elements of the symmetric matrix  $\Phi$  are computed by:

$$\Phi(i, j) = \sum_{n=j}^{39} H(n-i)H(n-j), \quad (j \geq i).$$

The correlation in the numerator is given by:

$$C = \sum_{i=0}^{N_p-1} \vartheta_i d(m_i),$$

where  $m_i$  is the position of the  $i$ th pulse and  $\vartheta_i$  is its amplitude. For the complexity reason, all

the amplitudes  $\{\vartheta_i\}$  are set to +1 or -1; that is,

$$\vartheta_i = \text{SIGN}(i), \quad i = n_p = 0, \dots, N_p - 1.$$

The energy in the denominator is given by:

$$E_D = \sum_{i=0}^{N_p-1} \phi(m_i, m_i) + 2 \sum_{i=0}^{N_p-2} \sum_{j=i+1}^{N_p-1} \vartheta_i \vartheta_j \phi(m_i, m_j).$$

To simplify the search procedure, the pulse signs are preset by using the signal  $x_2(n)$ ,

which is a weighted sum of the normalized  $d(n)$  vector and the normalized target signal of  $x_2(n)$

in the residual domain  $res_2(n)$ :

$$b(n) = \frac{res_2(n)}{\sqrt{\sum_{i=0}^{39} res_2(i)res_2(i)}} + \frac{2d(n)}{\sqrt{\sum_{i=0}^{39} d(i)d(i)}}, \quad n=0, 1, \dots, 39$$

If the sign of the  $i$ th ( $i=n_p$ ) pulse located at  $m_i$  is encoded, it is set to the sign of signal  $b(n)$  at that position, i.e.,  $\text{SIGN}(i) = \text{sign}[b(m_i)]$ .

In the present embodiment, the fixed codebook 261 has 2 or 3 subcodebooks for each of the encoding bit rates. Of course many more might be used in other embodiments. Even with several subcodebooks, however, the searching of the fixed codebook 261 is very fast using the following procedure. In a first searching turn, the encoder processing circuitry searches the pulse positions sequentially from the first pulse ( $n_p=0$ ) to the last pulse ( $n_p=N_p-1$ ) by considering the influence of all the existing pulses.

In a second searching turn, the encoder processing circuitry corrects each pulse position sequentially from the first pulse to the last pulse by checking the criterion value  $A_k$  contributed from all the pulses for all possible locations of the current pulse. In a third turn, the functionality of the second searching turn is repeated a final time. Of course further turns may be utilized if the added complexity is not prohibitive.

The above searching approach proves very efficient, because only one position of one pulse is changed leading to changes in only one term in the criterion numerator  $C$  and few terms in the criterion denominator  $E_D$  for each computation of the  $A_k$ . As an example, suppose a pulse subcodebook is constructed with 4 pulses and 3 bits per pulse to encode the position. Only 96 ( $4 \text{ pulses} \times 2^3 \text{ positions per pulse} \times 3 \text{ turns} = 96$ ) simplified computations of the criterion  $A_k$  need be performed.

Moreover, to save the complexity, usually one of the subcodebooks in the fixed codebook 261 is chosen after finishing the first searching turn. Further searching turns are done only with the chosen subcodebook. In other embodiments, one of the subcodebooks might be chosen only after the second searching turn or thereafter should processing resources so permit.

The Gaussian codebook is structured to reduce the storage requirement and the computational complexity. A comb-structure with two basis vectors is used. In the comb-

structure, the basis vectors are orthogonal, facilitating a low complexity search. In the AMR coder, the first basis vector occupies the even sample positions, (0,2,...,38), and the second basis vector occupies the odd sample positions, (1,3,...,39).

The same codebook is used for both basis vectors, and the length of the codebook vectors is 20 samples (half the subframe size).

All rates (6.65, 5.8 and 4.55 kbps) use the same Gaussian codebook. The Gaussian codebook,  $CB_{Gauss}$ , has only 10 entries, and thus the storage requirement is  $10 \cdot 20 = 200$  16-bit words. From the 10 entries, as many as 32 code vectors are generated. An index,  $idx_g$ , to one basis vector 22 populates the corresponding part of a code vector,  $c_{idx_g}$ , in the following way:

$$c_{idx_g}(2 \cdot (i - \tau) + \delta) = CB_{Gauss}(l, i) \quad i = \tau, \tau + 1, \dots, 19$$

$$c_{idx_g}(2 \cdot (i + 20 - \tau) + \delta) = CB_{Gauss}(l, i) \quad i = 0, 1, \dots, \tau - 1$$

where the table entry,  $l$ , and the shift,  $\tau$ , are calculated from the index,  $idx_g$ , according to:

$$\tau = \text{trunc}\{idx_g / 10\}$$

$$l = idx_g - 10 \cdot \tau$$

and  $\delta$  is 0 for the first basis vector and 1 for the second basis vector. In addition, a sign is applied to each basis vector.

Basically, each entry in the Gaussian table can produce as many as 20 unique vectors, all with the same energy due to the circular shift. The 10 entries are all normalized to have identical energy of 0.5, i.e.,

$$\sum_{i=0}^{19} CB_{Gauss}(l, i)^2 = 0.5, \quad l = 0, 1, \dots, 9$$

That means that when both basis vectors have been selected, the combined code vector,  $c_{idx_g, idx_l}$ , will have unity energy, and thus the final excitation vector from the Gaussian subcodebook will

have unity energy since no pitch enhancement is applied to candidate vectors from the Gaussian subcodebook.

The search of the Gaussian codebook utilizes the structure of the codebook to facilitate a low complexity search. Initially, the candidates for the two basis vectors are searched independently based on the ideal excitation,  $res_j$ . For each basis vector, the two best candidates, along with the respective signs, are found according to the mean squared error. This is exemplified by the equations to find the best candidate, index  $idx_g$ , and its sign,  $s_{idx_g}$ :

$$idx_g = \max_{i=0,1,\dots,N_{Gauss}-1} \left\{ \sum_{j=0}^{19} res_j(2 \cdot i + \delta) \cdot c_{idx_g}(2 \cdot i + \delta) \right\}$$

$$s_{idx_g} = \text{sign} \left( \sum_{j=0}^{19} res_j(2 \cdot i + \delta) \cdot c_{idx_g}(2 \cdot i + \delta) \right)$$

where  $N_{Gauss}$  is the number of candidate entries for the basis vector. The remaining parameters are explained above. The total number of entries in the Gaussian codebook is  $2 \cdot 2 \cdot N_{Gauss}$ . The fine search minimizes the error between the weighted speech and the weighted synthesized speech considering the possible combination of candidates for the two basis vectors from the pre-selection. If  $c_{k_0, k_1}$  is the Gaussian code vector from the candidate vectors represented by the indices  $k_0$  and  $k_1$  and the respective signs for the two basis vectors, then the final Gaussian code vector is selected by maximizing the term:

$$A_{k_0, k_1} = \frac{(C_{k_0, k_1})^T (d^T \cdot C_{k_0, k_1})^2}{E_{D_{k_0, k_1}} \cdot C_{k_0, k_1}^T \cdot C_{k_0, k_1}}$$

over the candidate vectors.  $d = H^T x_s$  is the correlation between the target signal  $x_s(n)$  and the impulse response  $h(n)$  (without the pitch enhancement), and  $H$  is a lower triangular Toeplitz

convolution matrix with diagonal  $h(0)$  and lower diagonals  $h(1), \dots, h(39)$ , and  $\Phi = H' H$  is the matrix of correlations of  $h(n)$ .

More particularly, in the present embodiment, two subcodebooks are included (or utilized) in the fixed codebook 261 with 31 bits in the 11 kbps encoding mode. In the first subcodebook, the innovation vector contains 8 pulses. Each pulse has 3 bits to code the pulse position. The signs of 6 pulses are transmitted to the decoder with 6 bits. The second subcodebook contains innovation vectors comprising 10 pulses. Two bits for each pulse are assigned to code the pulse position which is limited in one of the 10 segments. Ten bits are spent for 10 signs of the 10 pulses. The bit allocation for the subcodebooks used in the fixed codebook 261 can be summarized as follows:

*Subcodebook1: 8 pulses X 3 bits/pulse + 6 signs = 30 bits*  
*Subcodebook2: 10 pulses X 2 bits/pulse + 10 signs = 30 bits*

One of the two subcodebooks is chosen at the block 275 (Fig. 2) by favoring the second subcodebook using adaptive weighting applied when comparing the criterion value  $F1$  from the first subcodebook to the criterion value  $F2$  from the second subcodebook:

*if ( $W_c \cdot F1 > F2$ ), the first subcodebook is chosen,*  
*else, the second subcodebook is chosen,*

where the weighting,  $0 < W_c \leq 1$ , is defined as:

$$W_c = \begin{cases} 1.0, & \text{if } P_{NSR} < 0.5, \\ 1.0 - 0.3 P_{NSR} (1.0 - 0.5 R_p), & \text{min } (P_{sharp} + 0.5, 1.0), \end{cases}$$

$P_{NSR}$  is the background noise to speech signal ratio (i.e., the "noise level" in the block 279),  $R_p$  is the normalized LTP gain, and  $P_{sharp}$  is the sharpness parameter of the ideal excitation  $res_2(n)$  (i.e., the "sharpness" in the block 279).

In the 8 kbps mode, two subcodebooks are included in the fixed codebook 261 with 20 bits. In the first subcodebook, the innovation vector contains 4 pulses. Each pulse has 4 bits to code the pulse position. The signs of 3 pulses are transmitted to the decoder with 3 bits. The second subcodebook contains innovation vectors having 10 pulses. One bit for each of 9 pulses is assigned to code the pulse position which is limited in one of the 10 segments. Ten bits are spent for 10 signs of the 10 pulses. The bit allocation for the subcodebook can be summarized as the following:

*Subcodebook1: 4 pulses X 4 bits/pulse + 3 signs = 19 bits*  
*Subcodebook2: 9 pulses X 1 bits/pulse + 1 pulse X 0 bit + 10 signs = 19 bits*

One of the two subcodebooks is chosen by favoring the second subcodebook using adaptive weighting applied when comparing the criterion value  $F1$  from the first subcodebook to the criterion value  $F2$  from the second subcodebook as in the 11 kbps mode. The weighting,  $0 < W_c \leq 1$ , is defined as:

$$W_c = 1.0 - 0.6 P_{NSR} (1.0 - 0.5 R_p), \text{ min } (P_{sharp} + 0.5, 1.0).$$

The 6.65 kbps mode operates using the long-term preprocessing (PP) or the traditional LTP. A pulse subcodebook of 18 bits is used when in the PP-mode. A total of 13 bits are allocated for three subcodebooks when operating in the LTP-mode. The bit allocation for the subcodebooks can be summarized as follows:

*PP-mode:*

*Subcodebook: 5 pulses X 3 bits/pulse + 3 signs = 18 bits*

*LTP-mode:*

*Subcodebook1: 3 pulses X 3 bits/pulse + 3 signs = 12 bits, phase\_mode=1,*  
*Subcodebook2: 3 pulses X 3 bits/pulse + 2 signs = 11 bits, phase\_mode=0,*  
*Subcodebook3: Gaussian subcodebook of 11 bits.*

One of the 3 subcodebooks is chosen by favoring the Gaussian subcodebook when searching with LTP-mode. Adaptive weighting is applied when comparing the criterion value from the

two pulse subcodebooks to the criterion value from the Gaussian subcodebook. The weighting,

$0 < W_c \leq 1$ , is defined as:

$$W_c = 1.0 - 0.9 P_{nsr} (1.0 - 0.5 R_p) \cdot \min(P_{hmp} + 0.5, 1.0),$$

if (noise - like unvoiced),  $W_c \Leftarrow W_c \cdot (0.2 R_p (1.0 - P_{hmp}) + 0.8)$ .

The 5.8 kbps encoding mode works only with the long-term preprocessing (PP). Total 14

bits are allocated for three subcodebooks. The bit allocation for the subcodebooks can be

summarized as the following:

*Subcodebook1: 4 pulses X 3 bits/pulse + 1 signs = 13 bits, phase\_mode=1,*

*Subcodebook2: 3 pulses X 3 bits/pulse + 3 signs = 12 bits, phase\_mode=0,*

*Subcodebook3: Gaussian subcodebook of 12 bits.*

One of the 3 subcodebooks is chosen favoring the Gaussian subcodebook with aaptive weighting

applied when comparing the criterion value from the two pulse subcodebooks to the criterion

value from the Gaussian subcodebook. The weighting,  $0 < W_c \leq 1$ , is defined as:

$$W_c = 1.0 - P_{nsr} (1.0 - 0.5 R_p) \cdot \min(P_{hmp} + 0.6, 1.0),$$

if (noise - like unvoiced),  $W_c \Leftarrow W_c \cdot (0.3 R_p (1.0 - P_{hmp}) + 0.7)$ .

The 4.55 kbps bit rate mode works only with the long-term preprocessing (PP). Total 10

bits are allocated for three subcodebooks. The bit allocation for the subcodebooks can be

summarized as the following:

*Subcodebook1: 2 pulses X 4 bits/pulse + 1 signs = 9 bits, phase\_mode=1,*

*Subcodebook2: 2 pulses X 3 bits/pulse + 2 signs = 8 bits, phase\_mode=0,*

*Subcodebook3: Gaussian subcodebook of 8 bits.*

One of the 3 subcodebooks is chosen by favoring the Gaussian subcodebook with weighting

applied when comparing the criterion value from the two pulse subcodebooks to the criterion

value from the Gaussian subcodebook. The weighting,  $0 < W_c \leq 1$ , is defined as:

$$W_c = 1.0 - 1.2 P_{nsr} (1.0 - 0.5 R_p) \cdot \min(P_{hmp} + 0.6, 1.0),$$

if (noise - like unvoiced),  $W_c \Leftarrow W_c \cdot (0.6 R_p (1.0 - P_{hmp}) + 0.4)$ .

For 4.55, 5.8, 6.65 and 8.0 kbps bit rate encoding modes, a gain re-optimization

procedure is performed to jointly optimize the adaptive and fixed codebook gains,  $g_p$  and  $g_c$ .

respectively, as indicated in Fig. 3. The optimal gains are obtained from the following

correlations given by:

$$g_p = \frac{R_1 R_2 - R_3 R_4}{R_3 R_2 - R_1 R_4}$$

$$g_c = \frac{R_1 - g_p R_2}{R_3}$$

where  $R_1 \Leftarrow \bar{C}_p, \bar{T}_p >$ ,  $R_2 \Leftarrow \bar{C}_c, \bar{C}_c >$ ,  $R_3 \Leftarrow \bar{C}_p, \bar{C}_c >$ ,  $R_4 \Leftarrow \bar{C}_c, \bar{T}_p >$ , and

$R_5 \Leftarrow \bar{C}_p, \bar{C}_c >$ ,  $\bar{C}_c$ ,  $\bar{C}_p$ , and  $\bar{T}_p$  are filtered fixed codebook excitation, filtered adaptive codebook excitation and the target signal for the adaptive codebook search.

For 11 kbps bit rate encoding, the adaptive codebook gain,  $g_p$ , remains the same as that

computed in the close-loop pitch search. The fixed codebook gain,  $g_c$ , is obtained as:

$$g_c = \frac{R_5}{R_1}$$

where  $R_5 \Leftarrow \bar{C}_c, \bar{T}_p >$  and  $\bar{T}_p = \bar{T}_p - g_p \bar{C}_p$ .

Original CELP algorithm is based on the concept of analysis by synthesis (waveform matching). At low bit rate or when coding noisy speech, the waveform matching becomes difficult so that the gains are up-down, frequently resulting in unnatural sounds. To compensate for this problem, the gains obtained in the analysis by synthesis close-loop sometimes need to be modified or normalized.

There are two basic gain normalization approaches. One is called open-loop approach which normalizes the energy of the synthesized excitation to the energy of the unquantized residual signal. Another one is close-loop approach with which the normalization is done considering the perceptual weighting. The gain normalization factor is a linear combination of the one from the close-loop approach and the one from the open-loop approach; the weighting coefficients used for the combination are controlled according to the LPC gain.

The decision to do the gain normalization is made if one of the following conditions is met: (a) the bit rate is 8.0 or 6.65 kbps, and noise-like unvoiced speech is true; (b) the noise level  $P_{NSR}$  is larger than 0.5; (c) the bit rate is 6.65 kbps, and the noise level  $P_{NSR}$  is larger than 0.2; and (d) the bit rate is 5.8 or 4.45 kbps.

The residual energy,  $E_{res}$ , and the target signal energy,  $E_{TSP}$ , are defined respectively as:

$$E_{res} = \sum_{n=0}^{L_{SF}-1} res^2(n)$$

$$E_{TSP} = \sum_{n=0}^{L_{SF}-1} T_p^2(n)$$

Then the smoothed open-loop energy and the smoothed closed-loop energy are evaluated by:

$$\begin{aligned} & \text{if (first subframe is true)} \\ & \quad OL\_Eg = E_{res} \\ & \text{else} \\ & \quad OL\_Eg \Leftarrow \beta_{sub} \cdot OL\_Eg + (1 - \beta_{sub}) E_{res} \\ & \text{if (first subframe is true)} \\ & \quad CL\_Eg = E_{TSP} \\ & \text{else} \\ & \quad CL\_Eg \Leftarrow \beta_{sub} \cdot CL\_Eg + (1 - \beta_{sub}) E_{TSP} \end{aligned}$$

where  $\beta_{sub}$  is the smoothing coefficient which is determined according to the classification. After having the reference energy, the open-loop gain normalization factor is calculated:

$$OL\_g = MIN \left( C_{ol} \sqrt{\frac{OL\_Eg}{L_{SF-1}}}, \frac{1.2}{g_p} \right)$$

where  $C_{ol}$  is 0.8 for the bit rate 11.0 kbps, for the other rates  $C_{ol}$  is 0.7, and  $v(n)$  is the excitation:  $v(n) = v_o(n)g_p + v_c(n)g_c$ ,  $n=0,1,\dots,L_{SF}-1$ .

where  $g_p$  and  $g_c$  are unquantized gains. Similarly, the closed-loop gain normalization factor is:

$$CL\_g = MIN \left( C_{cl} \sqrt{\frac{CL\_Eg}{L_{SF-1}}}, \frac{1.2}{g_p} \right)$$

where  $C_{cl}$  is 0.9 for the bit rate 11.0 kbps, for the other rates  $C_{cl}$  is 0.8, and  $y(n)$  is the filtered signal ( $y(n)=v(n)*h(n)$ ):

$$y(n) = y_o(n)g_p + y_c(n)g_c, \quad n=0,1,\dots,L_{SF}-1.$$

The final gain normalization factor,  $g_f$ , is a combination of  $CL\_g$  and  $OL\_g$ , controlled in terms of an LPC gain parameter,  $C_{LPC}$ ,

if (speech is true or the rate is 11 kbps)

$$g_f = C_{LPC} OL\_g + (1 - C_{LPC}) CL\_g$$

$$g_f = MAX(1.0, g_f)$$

$$g_f = MIN(g_f, 1 + C_{LPC})$$

if (background noise is true and the rate is smaller than 11 kbps)

$$g_f = 1.2 \cdot MIN(CL\_g, OL\_g)$$

where  $C_{LPC}$  is defined as:

$$C_{LPC} = MIN(\sqrt{E_{res}/E_{TSP}}, 0.8)/0.8$$

Once the gain normalization factor is determined, the unquantized gains are modified:

$$g_p \leftarrow g_p \cdot g_f$$

For 4.55, 5.8, 6.65 and 8.0 kbps bit rate encoding, the adaptive codebook gain and the fixed codebook gain are vector quantized using 6 bits for rate 4.55 kbps and 7 bits for the other rates. The gain codebook search is done by minimizing the mean squared weighted error,  $Err$ , between the original and reconstructed speech signals:

$$Err = \left\| \vec{r}_n - g_p \vec{C}_p - g_c \vec{C}_c \right\|^2$$

For rate 11.0 kbps, scalar quantization is performed to quantize both the adaptive codebook gain,  $g_p$ , using 4 bits and the fixed codebook gain,  $g_c$ , using 5 bits each.

The fixed codebook gain,  $g_c$ , is obtained by MA prediction of the energy of the scaled fixed codebook excitation in the following manner. Let  $E(n)$  be the mean removed energy of the scaled fixed codebook excitation in (dB) at subframe  $n$  be given by:

$$E(n) = 10 \log \left( \frac{1}{40} g_c^2 \sum_{i=0}^{39} c^2(i) \right) - \bar{E}$$

where  $c(i)$  is the unscaled fixed codebook excitation, and  $\bar{E} = 30$  dB is the mean energy of scaled fixed codebook excitation.

The predicted energy is given by:

$$\bar{E}(n) = \sum_{i=1}^4 b_i \hat{R}(n-i)$$

where  $[b_1, b_2, b_3, b_4] = [0.68, 0.58, 0.34, 0.19]$  are the MA prediction coefficients and  $\hat{R}(n)$  is the quantized prediction error at subframe  $n$ .

The predicted energy is used to compute a predicted fixed codebook gain  $g_c$  (by substituting  $E(n)$  by  $\bar{E}(n)$  and  $g_c$  by  $g_c$ ). This is done as follows. First, the mean energy of the unscaled fixed codebook excitation is computed as:

$$E_1 = 10 \log \left( \frac{1}{40} \sum_{i=0}^{39} c^2(i) \right)$$

and then the predicted gain  $g_c$  is obtained as:

$$g_c = 10^{(0.05(\bar{E}(n) + \bar{E} - E_1))}$$

A correction factor between the gain,  $g_c$ , and the estimated one,  $g_c$ , is given by:

$$\gamma = \frac{g_c}{g_c}$$

It is also related to the prediction error as:

$$R(n) = E(n) - \bar{E}(n) = 20 \log \gamma$$

The codebook search for 4.55, 5.8, 6.65 and 8.0 kbps encoding bit rates consists of two

steps. In the first step, a binary search of a single entry table representing the quantized prediction error is performed. In the second step, the index  $Index\_1$  of the optimum entry that is closest to the unquantized prediction error in mean square error sense is used to limit the search of the two-dimensional VQ table representing the adaptive codebook gain and the prediction error. Taking advantage of the particular arrangement and ordering of the VQ table, a fast search using few candidates around the entry pointed by  $Index\_1$  is performed. In fact, only about half of the VQ table entries are tested to lead to the optimum entry with  $Index\_2$ . Only  $Index\_2$  is transmitted.

For 11.0 kbps bit rate encoding mode, a full search of both scalar gain codebooks are

used to quantize  $g_p$  and  $g_c$ . For  $g_p$ , the search is performed by minimizing the error

$$Err = abs(g_p - \bar{g}_p). \text{ Whereas for } g_c, \text{ the search is performed by minimizing the error}$$

$$Err = \|\bar{T}_p - \bar{g}_p \bar{C}_p - g_c \bar{C}_c\|^2.$$

An update of the states of the synthesis and weighting filters is needed in order to

compute the target signal for the next subframe. After the two gains are quantized, the excitation signal,  $u(n)$ , in the present subframe is computed as:

$$u(n) = \bar{g}_p v(n) + \bar{g}_c c(n), n = 0, 39,$$

where  $\bar{g}_p$  and  $\bar{g}_c$  are the quantized adaptive and fixed codebook gains respectively,  $v(n)$  the adaptive codebook excitation (interpolated past excitation), and  $c(n)$  is the fixed codebook excitation. The state of the filters can be updated by filtering the signal  $r(n) - u(n)$  through the filters  $1/\bar{A}(z)$  and  $W(z)$  for the 40-sample subframe and saving the states of the filters. This would normally require 3 filterings.

A simpler approach which requires only one filtering is as follows. The local synthesized speech at the encoder,  $\hat{s}(n)$ , is computed by filtering the excitation signal through  $1/\bar{A}(z)$ . The output of the filter due to the input  $r(n) - u(n)$  is equivalent to  $e(n) = s(n) - \hat{s}(n)$ , so the states of the synthesis filter  $1/\bar{A}(z)$  are given by  $e(n)$ ,  $n = 0, 39$ . Updating the states of the filter  $W(z)$  can be done by filtering the error signal  $e(n)$  through this filter to find the perceptually weighted error  $e_w(n)$ . However, the signal  $e_w(n)$  can be equivalently found by:

$$e_w(n) = T_p(n) - \bar{g}_p C_p(n) - \bar{g}_c C_c(n).$$

The states of the weighting filter are updated by computing  $e_w(n)$  for  $n = 30$  to 39.

The function of the decoder consists of decoding the transmitted parameters (dLP parameters, adaptive codebook vector and its gain, fixed codebook vector and its gain) and performing synthesis to obtain the reconstructed speech. The reconstructed speech is then postfiltered and upsampled.

The decoding process is performed in the following order. First, the LP filter parameters are encoded. The received indices of LSF quantization are used to reconstruct the quantized LSF vector. Interpolation is performed to obtain 4 interpolated LSF vectors (corresponding to 4 subframes). For each subframe, the interpolated LSF vector is converted to LP filter coefficient domain,  $a_1$ , which is used for synthesizing the reconstructed speech in the subframe.

For rates 4.55, 5.8 and 6.65 (during PP\_mode) kbps bit rate encoding modes, the received pitch index is used to interpolate the pitch lag across the entire subframe. The following three steps are repeated for each subframe:

- 1) Decoding of the gains: for bit rates of 4.55, 5.8, 6.65 and 8.0 kbps, the received index is used to find the quantized adaptive codebook gain,  $\bar{g}_p$ , from the 2-dimensional VQ table. The same index is used to get the fixed codebook gain correction factor  $\bar{\gamma}$  from the same quantization table. The quantized fixed codebook gain,  $\bar{g}_c$ , is obtained following these steps:

• the predicted energy is computed  $\bar{E}(n) = \sum_{i=1}^4 b_i \hat{R}(n-i)$ ;

- the energy of the unscaled fixed codebook excitation is calculated

$$\text{as } E_i = 10 \log \left( \frac{1}{40} \sum_{i=0}^{39} c^2(i) \right); \text{ and}$$

- the predicted gain  $\hat{g}_e$  is obtained as  $\hat{g}_e = 10^{(0.05(\bar{E}(n) - \bar{E} - E_e))}$ .

The quantized fixed codebook gain is given as  $\hat{g}_e = \hat{g}_e$ . For 11 kbps bit rate, the received adaptive codebook gain index is used to readily find the quantized adaptive gain.

$\hat{g}_e$  from the quantization table. The received fixed codebook gain index gives the fixed codebook gain correction factor  $\gamma$ . The calculation of the quantized fixed codebook gain,  $\hat{g}_e$ , follows the same steps as the other rates.

- 2) Decoding of adaptive codebook vector: for 8.0, 11.0 and 6.65 (during LTP\_mode=1) kbps bit rate encoding modes, the received pitch index (adaptive codebook index) is used to find the integer and fractional parts of the pitch lag. The adaptive codebook  $v(n)$  is found by interpolating the past excitation  $u(n)$  (at the pitch delay) using the FIR filters.

- 3) Decoding of fixed codebook vector: the received codebook indices are used to extract the type of the codebook (pulse or Gaussian) and either the amplitudes and positions of the excitation pulses or the bases and signs of the Gaussian excitation. In either case, the reconstructed fixed codebook excitation is given as  $c(n)$ . If the integer part of the pitch lag is less than the subframe size 40 and the chosen excitation is pulse type, the pitch sharpening is applied. This translates into modifying  $c(n)$  as  $c(n) = c(n) + \beta c(n - T)$ , where  $\beta$  is the decoded pitch gain  $\hat{g}_p$  from the previous subframe bounded by [0.2, 1.0].

The excitation at the input of the synthesis filter is given by

$$u(n) = \hat{g}_p v(n) + \hat{g}_e c(n), n = 0, 39. \text{ Before the speech synthesis, a post-processing of the}$$

excitation elements is performed. This means that the total excitation is modified by emphasizing the contribution of the adaptive codebook vector:

$$\bar{u}(n) = \begin{cases} u(n) + 0.25\beta\hat{g}_p v(n), & \hat{g}_p > 0.5 \\ u(n), & \hat{g}_p \leq 0.5 \end{cases}$$

Adaptive gain control (AGC) is used to compensate for the gain difference between the unemphasized excitation  $u(n)$  and emphasized excitation  $\bar{u}(n)$ . The gain scaling factor  $\eta$  for the emphasized excitation is computed by:

$$\eta = \begin{cases} \frac{\sum_{n=0}^{39} u^2(n)}{\sum_{n=0}^{39} \bar{u}^2(n)}, & \hat{g}_p > 0.5 \\ 1.0, & \hat{g}_p \leq 0.5 \end{cases}$$

The gain-scaled emphasized excitation  $\bar{u}(n)$  is given by:

$$\bar{u}(n) = \eta \bar{u}(n).$$

The reconstructed speech is given by:

$$\bar{s}(n) = \bar{u}(n) - \sum_{i=1}^{10} \bar{a}_i \bar{s}(n-i), n = 0 \text{ to } 39,$$

where  $\bar{a}_i$  are the interpolated LP filter coefficients. The synthesized speech  $\bar{s}(n)$  is then passed through an adaptive postfilter.

Post-processing consists of two functions: adaptive postfiltering and signal up-scaling.

The adaptive postfilter is the cascade of three filters: a formant postfilter and two tilt compensation filters. The postfilter is updated every subframe of 5 ms. The formant postfilter is given by:

$$H_f(z) = \frac{\bar{A}(z/\gamma_a)}{\bar{A}(z/\gamma_a)}$$



where  $\bar{A}(z)$  is the received quantized and interpolated LP inverse filter and  $\gamma_a$  and  $\gamma_f$  control the amount of the formant postfiltering.

The first tilt compensation filter  $H_n(z)$  compensates for the tilt in the formant postfilter

$H_f(z)$  and is given by:

$$H_n(z) = (1 - \mu z^{-1})$$

where  $\mu = \gamma_n k_1$  is a tilt factor, with  $k_1$  being the first reflection coefficient calculated on the truncated impulse response  $h_f(n)$ , of the formant postfilter  $k_1 = \frac{r_1(1)}{r_1(0)}$  with:

$$r_n(i) = \sum_{j=0}^{L_n-i-1} h_f(j)h_f(j+i), (L_n = 22).$$

The postfiltering process is performed as follows. First, the synthesized speech  $\bar{x}(n)$  is inverse filtered through  $\bar{A}(z/\gamma_a)$  to produce the residual signal  $\bar{r}(n)$ . The signal  $\bar{r}(n)$  is filtered by the synthesis filter  $1/\bar{A}(z/\gamma_a)$  is passed to the first tilt compensation filter  $h_n(z)$  resulting in the postfiltered speech signal  $\bar{s}_f(n)$ .

Adaptive gain control (AGC) is used to compensate for the gain difference between the synthesized speech signal  $\bar{x}(n)$  and the postfiltered signal  $\bar{s}_f(n)$ . The gain scaling factor  $\gamma$  for the present subframe is computed by:

$$\gamma = \sqrt{\frac{\sum_{n=0}^{10} \bar{s}_f^2(n)}{\sum_{n=0}^{10} \bar{x}^2(n)}}$$

The gain-scaled postfiltered signal  $\bar{s}(n)$  is given by:

$$\bar{s}(n) = \beta(n)\bar{s}_f(n)$$

where  $\beta(n)$  is updated in sample by sample basis and given by:

$$\beta(n) = \alpha\beta(n-1) + (1-\alpha)\gamma$$

where  $\alpha$  is an AGC factor with value 0.9. Finally, up-scaling consists of multiplying the postfiltered speech by a factor 2 to undo the down scaling by 2 which is applied to the input signal.

Figs. 6 and 7 are drawings of an alternate embodiment of a 4 kbps speech codec that also illustrates various aspects of the present invention. In particular, Fig. 6 is a block diagram of a speech encoder 601 that is built in accordance with the present invention. The speech encoder 601 is based on the analysis-by-synthesis principle. To achieve toll quality at 4 kbps, the speech encoder 601 departs from the strict waveform-matching criterion of regular CELP coders and strives to catch the perceptual important features of the input signal.

The speech encoder 601 operates on a frame size of 20 ms with three subframes (two of 6.625 ms and one of 6.75 ms). A look-ahead of 15 ms is used. The one-way coding delay of the codec adds up to 55 ms.

At a block 615, the spectral envelope is represented by a 10<sup>th</sup> order LPC analysis for each frame. The prediction coefficients are transformed to the Line Spectrum Frequencies (LSFs) for quantization. The input signal is modified to better fit the coding model without loss of quality. This processing is denoted "signal modification" as indicated by a block 621. In order to improve the quality of the reconstructed signal, perceptual important features are estimated and emphasized during encoding.

The excitation signal for an LPC synthesis filter 625 is build from the two traditional components: 1) the pitch contribution; and 2) the innovation contribution. The pitch contribution is provided through use of an adaptive codebook 627. An innovation codebook 629 has several

subcodebooks in order to provide robustness against a wide range of input signals. To each of the two contributions a gain is applied which, multiplied with their respective codebook vectors and summed, provide the excitation signal.

The LSFs and pitch lag are coded on a frame basis, and the remaining parameters (the innovation codebook index, the pitch gain, and the innovation codebook gain) are coded for every subframe. The LSF vector is coded using predictive vector quantization. The pitch lag has an integer part and a fractional part constituting the pitch period. The quantized pitch period has a non-uniform resolution with higher density of quantized values at lower delays. The bit allocation for the parameters is shown in the following table.

**Table of Bit Allocation**

Parameter	Bits per 20 ms
LSFs	21
Pitch lag (adaptive codebook)	8
Gains	12
Innovation codebook	$3 \times 13 = 39$
Total	80

When the quantization of all parameters for a frame is complete the indices are multiplexed to form the 80 bits for the serial bit-stream.

Fig. 7 is a block diagram of a decoder 701 with corresponding functionality to that of the encoder of Fig. 6. The decoder 701 receives the 80 bits on a frame basis from a demultiplexor 711. Upon receipt of the bits, the decoder 701 checks the sync-word for a bad frame indication, and decides whether the entire 80 bits should be disregarded and frame erasure concealment applied. If the frame is not declared a frame erasure, the 80 bits are mapped to the parameter indices of the codec, and the parameters are decoded from the indices using the inverse quantization schemes of the encoder of Fig. 6.

When the LSFs, pitch lag, pitch gains, innovation vectors, and gains for the innovation vectors are decoded, the excitation signal is reconstructed via a block 715. The output signal is synthesized by passing the reconstructed excitation signal through an LPC synthesis filter 721.

To enhance the perceptual quality of the reconstructed signal both short-term and long-term post-processing are applied at a block 731.

Regarding the bit allocation of the 4 kbps codec (as shown in the prior table), the LSFs and pitch lag are quantized with 21 and 8 bits per 20 ms, respectively. Although the three subframes are of different size the remaining bits are allocated evenly among them. Thus, the innovation vector is quantized with 13 bits per subframe. This adds up to a total of 80 bits per 20 ms, equivalent to 4 kbps.

The estimated complexity numbers for the proposed 4 kbps codec are listed in the following table. All numbers are under the assumption that the codec is implemented on commercially available 16-bit fixed point DSPs in full duplex mode. All storage numbers are under the assumption of 16-bit words, and the complexity estimates are based on the floating point C-source code of the codec.

**Table of Complexity Estimates**

Computational complexity	30 MIPS
Program and data ROM	18 kwords
RAM	3 kwords

The decoder 701 comprises decode processing circuitry that generally operates pursuant to software control. Similarly, the encoder 601 (Fig. 6) comprises encoder processing circuitry also operating pursuant to software control. Such processing circuitry may coexist, at least in part, within a single processing unit such as a single DSP.

Fig. 8a is a timing diagram of an exemplary pitch lag contour over two speech frames to which continuous warping techniques are applied in accordance with the present invention. In particular, an exemplary pitch lag contour, an original pitch lag contour 811, typically varies rather slowly over time. From a beginning of a first frame, as indicated by a marker 813, the original pitch lag contour 811 varies generally upward through a plurality of subframes, as indicated by subframe markers 819 and 821. Similarly, the upward trend can be seen in a second frame ending at a marker 811.

Without applying warping of the present invention, it can be appreciated that the amount of bits needed to code the original pitch lag contour 811 might prove excessive, especially at the lower encoder bit rates. Moreover, any attempt to search for a match of such pitch contour, such as shifting each of the pitch pulses in an original residual, proves difficult and requires reliable endpoint detection to maintain signal continuity.

Fig. 8b is a timing diagram illustrating a linear pitch contour to which continuous warping of the original pitch lag contour is applied in accordance with the present invention. Specifically, a linear segment 831 for a first frame, a linear segment 833 for a second frame, etc., provide a basis for warping the pitch lag contour 811. By performing continuous warping, the pitch contour 811 is effectively compressed during some periods, e.g., at a time period 835, and expanded during others, e.g., during a time period 837 to match the contour defined by the segments 831, 833, and so on.

From frame to frame such warping takes place, i.e., continuous warping is applied. Such processing or portions thereof might take place on subframe, multiple subframe, multiple frame basis, or other time period, for example. Similarly, although only three subframes are shown, more or less might be used with equal or unequal time period definition.

The warping to conform the pitch lag contour defined by the segments 831 and 833, for example, may be applied to the residual speech signal in an open loop approach. Alternatively, in some embodiments such as the specific embodiment described above in reference to Figs. 2-4, continuous warping is applied to the weighted speech signal (although the original speech signal might alternatively have been used) in a closed loop fashion. Searching for the best match can be performed rapidly by finding the optimal end of the original (weighted or residual) signal with a limited range to make the modified signal match the new pitch contour.

Fig. 8c is a diagram illustrating the use of the new pitch contour of Fig. 8b which can be represented by a lesser number of bits than the original pitch contour of Fig. 8a. A new pitch contour 841 comprising the linear segments 831 and 833 is defined by encoding the pitch lag at each segment marker. Having received such coding information, the decoder can reconstruct intermediate pitch lag values merely through interpolation, for example, as indicated at the subframe markers.

Fig. 9 is a flow diagram illustrating an embodiment of the continuous warping approach and an associated fast searching process used by an encoder of the present invention to carry out the functionality described in reference to Figs. 8a-c on a residual signal using an open loop approach. At a block 909, the encoder, i.e., the encoder processing circuitry operating pursuant to software instruction, first identifies maps the original residual to the modified residual, i.e., the original residual is mapped to a linear pitch contour defined by a previous and a current frame pitch lag value.

Specifically, at the block 909, the original residual having a  $T_{\text{start}}$  and a  $T_{\text{end}}$  is mapped to a modified residual defined by a  $T_{\text{start}}$  and a  $T_{\text{end}}$ . Thereafter, at a block 913, the encoder identifies a range in which an optimal value of  $T_{\text{end}}$  is searched. The search is performed at a

block 917 to make the modified residual best fit the pitch contour. With the optimal endpoint

$T_{end}$  found, at a block 921, the original residual is warped from the  $T_{start}$  and the optimal  $T_{end}$  to

the modified residual ( $T_{start}$  and  $T_{end}$ ) as follows:

$$T_{start} = T_{start} + L,$$

$$T_{start} = T_{start} + L (T_{end} - T_{start}) / (T_{end} - T_{start}),$$

where  $L$  comprises the working step size.

Fig. 10 is a flow diagram illustrating an alternate embodiment of functionality of a speech encoder of the present invention that performs continuous warping to the weighted speech signal in a closed loop approach. In particular, at a block 1011, the encoder estimates pitch lag at the end of a frame. Such estimation is based on the normalized correlation:

$$R_k = \frac{\sum_{n=0}^L s_w(n+n1)s_w(n+n1-k)}{\sqrt{\sum_{n=0}^L s_w^2(n+n1-k)}}$$

where  $s_w(n+n1)$ ,  $n=0,1,\dots,L-1$ , represents the last segment of the weighted speech signal including the look-ahead (the look-ahead length is 25 samples), and the size  $L$  is defined

according to the open-loop pitch lag  $T_{op}$  with the corresponding normalized correlation  $C_{T_{op}}$ :

$$\text{if } (C_{T_{op}} > 0.6)$$

$$L = \max(50, T_{op})$$

$$L = \min(80, L)$$

else

$$L = 80$$

To identify the pitch lag estimate, the encoder first selects one integer lag  $k$  maximizing

the  $R_k$  in the range  $k \in [T_{op} - 10, T_{op} + 10]$  bounded by [17, 145]. Then, the precise pitch lag  $P_m$

and the corresponding index  $I_m$  for the current frame is searched around the integer lag,  $[k-l,$

$k+l]$ , by up-sampling  $R_k$ . The possible candidates for the pitch lag are obtained from the table

named as *PriLagTab8b[i]*,  $i=0,1,\dots,127$ . Lastly, the pitch lag  $P_m = \text{PriLagTab8b}[I_m]$  is possibly

modified by checking the accumulated delay  $\tau_{acc}$  due to the modification of the speech signal:

$$\text{if } (\tau_{acc} > 5) \quad I_m \Leftarrow \min(I_m + 1, 127),$$

$$\text{if } (\tau_{acc} < -5) \quad I_m \Leftarrow \max(I_m - 1, 0);$$

it could be modified again:

$$\text{if } (\tau_{acc} > 10) \quad I_m \Leftarrow \min(I_m + 1, 127),$$

$$\text{if } (\tau_{acc} < -10) \quad I_m \Leftarrow \max(I_m - 1, 0);$$

The obtained index  $I_m$  will be sent to the decoder.

At a block 1013, the pitch lag contour,  $\tau_c(n)$ , is identified using both the current pitch

lag  $P_m$  and the previous pitch lag  $P_{m-1}$ :

$$\text{if } (|P_m - P_{m-1}| < 0.2 \min(P_m, P_{m-1}))$$

$$\tau_c(n) = P_{m-1} + n(P_m - P_{m-1}) / L_f, \quad n=0,1,\dots,L_f-1$$

$$\tau_c(n) = P_m, \quad n=L_f,\dots,170$$

else

$$\tau_c(n) = P_{m-1}, \quad n=0,1,\dots,39;$$

$$\tau_c(n) = P_m, \quad n=40,\dots,170$$

where  $L_f = 160$  is the frame size.

In the present embodiment, each frame is divided into 3 subframes for the long-term

preprocessing. For the first two subframes, the subframe size,  $L_n$ , is 53, and the subframe size for

searching,  $L_m$ , is 70. For the last subframe,  $L_p$  is 54 and  $L_{fp}$  is:

$$L_{fp} = \min(70, L_p + L_{md} - 10 - \tau_{acc}).$$

where  $L_{md}=25$  is the look-ahead and the maximum of the accumulated delay  $\tau_{acc}$  is limited to

14.

At a block 1015, the weighted speech signal is mapped to the pitch lag contour,  $\tau_c(n)$ .

In particular, the target for the modification process of the weighted speech, temporally

memorized in  $\{\hat{s}_w(m0+n), n=0,1,\dots,L_p-1\}$  is calculated by mapping, i.e., warping, the past modified weighted speech buffer,  $\hat{s}_w(m0+n), n<0$ , with the pitch lag contour,

$$\tau_c(n+m \cdot L_p), m=0,1,2,$$

$$\hat{s}_w(m0+n) = \sum_{i=-f_1}^f \hat{s}_w(m0+n-T_c(n)+i) I_f(i, T_c(n)), n=0,1,\dots,L_p-1,$$

where  $T_c(n)$  and  $T_d(n)$  are calculated by

$$\begin{aligned} T_c(n) &= \text{trunc}(\tau_c(n+m \cdot L_p)), \\ T_d(n) &= \tau_c(n) - T_c(n), \end{aligned}$$

$m$  is subframe number,  $I_f(i, T_c(n))$  is a set of interpolation coefficients, and  $f_1$  is 10. Then, the target for matching,  $\hat{s}_t(n), n=0,1,\dots,L_p-1$ , is calculated by weighting

$\hat{s}_w(m0+n), n=0,1,\dots,L_p-1$ , in the time domain:

$$\begin{aligned} \hat{s}_t(n) &= n \cdot \hat{s}_w(m0+n) / L_p, n=0,1,\dots,L_p-1, \\ \hat{s}_t(n) &= \hat{s}_w(m0+n), n=L_p,\dots,L_p-1. \end{aligned}$$

At a block 1017, the encoder calculates a relatively small shift range for seeking the best local delay. Specifically, the local integer shifting range  $[SR0, SR1]$  for searching for the best local delay is computed as the following:

if speech is unvoiced  
 $SR0=-1$ ,  
 $SR1=1$ .

else

$$\begin{aligned} SR0 &= \text{round}(-4 \min(1.0, \max(0.0, 1-0.4(P_{sh}-0.2)))) \\ SR1 &= \text{round}(4 \min(1.0, \max(0.0, 1-0.4(P_{sh}-0.2)))) \end{aligned}$$

where  $P_{sh} = \max(P_{sh1}, P_{sh2})$ ,  $P_{sh1}$  is the average to peak ratio (i.e., sharpness) from the target signal:

$$P_{sh1} = \frac{\sum_{n=0}^{L_p-1} |\hat{s}_w(m0+n)|}{L_p \max_{n=0,1,\dots,L_p-1} |\hat{s}_w(m0+n)|}$$

and  $P_{sh2}$  is the sharpness from the weighted speech signal,

$$P_{sh2} = \frac{\sum_{n=0}^{L_p-L_f/2-1} |\hat{s}_w(n+L_p/2)|}{(L_p-L_f/2) \max_{n=0,1,\dots,L_p-L_f/2-1} |\hat{s}_w(n+L_p/2)|}$$

where  $n0 = \text{trunc}(m0 + \tau_{acc} + 0.5)$  (here,  $m$  is subframe number and  $\tau_{acc}$  is the previous accumulated delay).

At a block 1019, the encoder searches for then adjusts the best local delay. Such searching involves use of linear time weighting. In particular, to find the best local delay,  $\tau_{opt}$ , at the end of the current processing subframe, a normalized correlation vector between the weighted speech signal and the modified matching target is defined as:

$$R_l(k) = \frac{\sum_{n=0}^{L_p-1} \hat{s}_w(n0+n+k) \hat{s}_t(n)}{\sqrt{\sum_{n=0}^{L_p-1} \hat{s}_w^2(n0+n+k) \sum_{n=0}^{L_p-1} \hat{s}_t^2(n)}}$$

A best local delay in the integer domain,  $k_{opt}$ , is selected by maximizing  $R_l(k)$  in the range of  $k \in [SR0, SR1]$ , which is corresponding to the real delay:

$$k_r = k_{opt} + n0 - m0 - \tau_{acc}$$

If  $R_l(k_{opt}) < 0.5$ ,  $k_r$  is set to zero.

In order to get a more precise local delay in the range  $[k_r-0.75+0.1j, j=0,1,\dots,15]$  around  $k_r$ ,  $R_l(k)$  is interpolated to obtain the fractional correlation vector,  $R_f(j)$ , which is given by:

$$R_f(j) = \sum_{i=-7}^8 R_l(k_{opt} + I_j + i) I_f(i, j), j=0,1,\dots,15,$$

where  $\{I_j(i,j)\}$  is a set of interpolation coefficients. The optimal fractional delay index,  $j_{opt}$ , is selected by maximizing  $R_d(j)$ . Finally, the best local delay,  $\tau_{opt}$ , at the end of the current processing subframe, is given:

$$\tau_{opt} = k_r - 0.75 + 0.1 j_{opt}$$

Once found, the best local delay is then adjusted as follows.

$$\tau_{opt} = \begin{cases} 0, & \text{if } \tau_{acc} + \tau_{opt} > 14 \\ \tau_{opt}, & \text{otherwise} \end{cases}$$

At a block 1021, the original weighted speech is warped from an original to a modified time region. Specifically, the modified weighted speech of the current subframe, memorized in  $\{\hat{s}_n(m0+n), n = 0, 1, \dots, L_2 - 1\}$  to update the buffer and produce the target for the fixed codebook search, is generated by warping the original weighted speech  $\{s_n(n)\}$  from the original time region:

$$[m0 + \tau_{acc}, m0 + \tau_{acc} + L_2 + \tau_{opt}],$$

to the modified time region.

$[m0, m0 + L_2]$ :

$$\hat{s}_n(m0+n) = \sum_{i=m-j_1+1}^{L_2} s_n(m0+n + \tau_w(n) + i) I_2(i, \tau_w(n)), \quad n = 0, 1, \dots, L_2 - 1,$$

where  $\tau_w(n)$  and  $T_w(n)$  are calculated by:

$$\begin{aligned} \tau_w(n) &= \text{trunc}(\tau_{acc} + n \cdot \tau_{opt} / L_2), \\ T_w(n) &= \tau_{acc} + n \cdot \tau_{opt} / L_2 - \tau_w(n). \end{aligned}$$

$\{I_2(i, \tau_w(n))\}$  is a set of interpolation coefficients.

To complete the process after having completed the warping of the weighted speech for the current subframe, the modified target weighted speech buffer is updated as follows:

$$\hat{s}_n(n) \Leftarrow \hat{s}_n(n + L_2), \quad n = 0, 1, \dots, n_m - 1.$$

The accumulated delay at the end of the current subframe is renewed by:

$$\tau_{acc} \Leftarrow \tau_{acc} + \tau_{opt}.$$

As previously articulated, although the continuous warping processes described with reference to Fig. 10 is applied to the weighted speech signal, it might alternatively be applied to the residual or, for example, to the original unweighted speech signal.

Of course, many other modifications and variations are also possible. In view of the above detailed description of the present invention and associated drawings, such other modifications and variations will now become apparent to those skilled in the art. It should also be apparent that such other modifications and variations may be effected without departing from the spirit and scope of the present invention.

In addition, the following Appendix A provides a list of many of the definitions, symbols and abbreviations used in this application. Appendices B and C respectively provide source and channel bit ordering information at various encoding bit rates used in one embodiment of the present invention. Appendices A, B and C comprise part of the detailed description of the present application, and, otherwise, are hereby incorporated herein by reference in its entirety.

## APPENDIX A

For purposes of this application, the following symbols, definitions and abbreviations apply.

adaptive codebook:	The adaptive codebook contains excitation vectors that are adapted for every subframe. The adaptive codebook is derived from the long term filter state. The pitch lag value can be viewed as an index into the adaptive codebook.	direct form coefficients:	One of the formats for storing the short term filter parameters. In the adaptive multi rate codec, all filters used to modify speech samples use direct form coefficients.
adaptive postfilter:	The adaptive postfilter is applied to the output of the short term synthesis filter to enhance the perceptual quality of the reconstructed speech. In the adaptive multi-rate codec (AMR), the adaptive postfilter is a cascade of two filters: a formant postfilter and a tilt compensation filter.	fixed codebook:	The fixed codebook contains excitation vectors for speech synthesis filters. The contents of the codebook are non-adaptive (i.e., fixed). In the adaptive multi rate codec, the fixed codebook for a specific rate is implemented using a multi-function codebook.
Adaptive Multi Rate codec:	The adaptive multi-rate code (AMR) is a speech and channel codec capable of operating at gross bit-rates of 11.4 kbps ("half-rate") and 22.8 kbs ("full-rate"). In addition, the codec may operate at various combinations of speech and channel coding (codec mode) bit-rates for each channel mode.	fractional lags:	A set of lag values having sub-sample resolution. In the adaptive multi rate codec a sub-sample resolution between $1/6^{\text{th}}$ and 1.0 of a sample is used.
AMR handover:	Handover between the full rate and half rate channel modes to optimize AMR operation.	full-rate (FR):	Full-rate channel or channel mode.
channel mode:	Half-rate (HR) or full-rate (FR) operation.	frame:	A time interval equal to 20 ms (160 samples at an 8 kHz sampling rate).
channel mode adaptation:	The control and selection of the (FR or HR) channel mode.	gross bit-rate:	The bit-rate of the channel mode selected (22.8 kbps or 11.4 kbps).
channel repacking:	Repacking of HR (and FR) radio channels of a given radio cell to achieve higher capacity within the cell.	half-rate (HR):	Half-rate channel or channel mode.
closed-loop pitch analysis:	This is the adaptive codebook search, i.e., a process of estimating the pitch (lag) value from the weighted input speech and the long term filter state. In the closed-loop search, the lag is searched using error minimization loop (analysis-by-synthesis). In the adaptive multi rate codec, closed-loop pitch search is performed for every subframe.	in-band signaling:	Signaling for DTX, Link Control, Channel and codec mode modification, etc. carried within the traffic.
codec mode:	For a given channel mode, the bit partitioning between the speech and channel codecs.	integer lags:	A set of lag values having whole sample resolution.
codec mode adaptation:	The control and selection of the codec mode bit-rates. Normally, implies no change to the channel mode.	interpolating filter:	An FIR filter used to produce an estimate of sub-sample resolution samples, given an input sampled with integer sample resolution.
		inverse filter:	This filter removes the short term correlation from the speech signal. The filter models an inverse frequency response of the vocal tract.
		lag:	The long term filter delay. This is typically the true pitch period, or its multiple or sub-multiple.
		Line Spectral Frequencies:	(see Line Spectral Pair)
		Line Spectral Pair:	Transformation of LPC parameters. Line Spectral Pairs are obtained by decomposing the inverse filter transfer function $A(z)$ to a set of two transfer functions, one having even symmetry and the other having odd symmetry. The Line Spectral Pairs (also called as Line Spectral Frequencies) are the roots of these polynomials on the z-unit circle).

**LP analysis window:**

For each frame, the short term filter coefficients are computed using the high pass filtered speech samples within the analysis window. In the adaptive multi rate codec, the length of the analysis window is always 240 samples. For each frame, two asymmetric windows are used to generate two sets of LP coefficient coefficients which are interpolated in the LSF domain to construct the perceptual weighting filter. Only a single set of LP coefficients per frame is quantized and transmitted to the decoder to obtain the synthesis filter. A lookahead of 25 samples is used for both HR and FR.

**LP coefficients:**

Linear Prediction (LP) coefficients (also referred as Linear Predictive Coding (LPC) coefficients) is a generic descriptive term for describing the short term filter coefficients.

**LTP Mode:**

Codec works with traditional LTP.

**mode:**

When used alone, refers to the source codec mode, i.e., to one of the source codecs employed in the AMR codec. (See also codec mode and channel mode.)

**multi-function codebook:**

A fixed codebook consisting of several subcodebooks constructed with different kinds of pulse innovation vector structures and noise innovation vectors, where codeword from the codebook is used to synthesize the excitation vectors.

**open-loop pitch search:**

A process of estimating the near optimal pitch lag directly from the weighted input speech. This is done to simplify the pitch analysis and confine the closed-loop pitch search to a small number of lags around the open-loop estimated lags. In the adaptive multi rate codec, open-loop pitch search is performed once per frame for PP mode and twice per frame for LTP mode.

**out-of-band signaling:**

Signaling on the GSM control channels to support link control.

**PP Mode:**

Codec works with pitch preprocessing.

**residual:**

The output signal resulting from an inverse filtering operation.

**short term synthesis filter:**

This filter introduces, into the excitation signal, short term correlation which models the impulse response of the vocal tract.

**perceptual weighting filter:**

This filter is employed in the analysis-by-synthesis search of the codebooks. The filter exploits the noise masking properties of the formants (vocal tract resonances) by weighting the error less in regions near the formant frequencies and more in regions away from them.

**subframe:**

A time interval equal to 5-10 ms (40-80 samples at an 8 KHz sampling rate).

**vector quantization:**

A method of grouping several parameters into a vector and quantizing them simultaneously.

**zero input response:**

The output of a filter due to past inputs, i.e. due to the present state of the filter, given that an input of zeros is applied.

**zero state response:**

The output of a filter due to the present input, given that no past inputs have been applied, i.e., given the state information in the filter is all zeroes.

$$A(z)$$

The inverse filter with unquantized coefficients

$$\hat{A}(z)$$

The inverse filter with quantized coefficients

$$H(z) = \frac{1}{\hat{A}(z)}$$

The speech synthesis filter with quantized coefficients

$$a_i$$

The unquantized linear prediction parameters (direct form coefficients)

$$\hat{a}_i$$

The quantized linear prediction parameters

$$\frac{1}{B(z)}$$

The long-term synthesis filter

$$W(z)$$

The perceptual weighting filter (unquantized coefficients)

$$\gamma_1, \gamma_2$$

The perceptual weighting factors

$$F_2(z)$$

Adaptive pre-filter

$$T$$

The nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe

$$\beta$$

The adaptive pre-filter coefficient (the quantized pitch gain)

$$H_f(z) = \frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)}$$

The formant postfilter

$$\gamma_n$$

Control coefficient for the amount of the formant post-filtering

$$\gamma_d$$

Control coefficient for the amount of the formant post-filtering



$H_f(z)$	Tilt compensation filter
$\gamma_r$	Control coefficient for the amount of the tilt compensation filtering
$\mu = \gamma_r k_1'$	A tilt factor, with $k_1'$ being the first reflection coefficient
$h_f(n)$	The truncated impulse response of the formant postfilter
$L_h$	The length of $h_f(n)$
$r_h(i)$	The auto-correlations of $h_f(n)$
$\hat{A}(z/\gamma_n)$	The inverse filter (numerator) part of the formant postfilter
$1/\hat{A}(z/\gamma_d)$	The synthesis filter (denominator) part of the formant postfilter
$\hat{r}(n)$	The residual signal of the inverse filter $\hat{A}(z/\gamma_n)$
$h_t(z)$	Impulse response of the tilt compensation filter
$\beta_{sc}(n)$	The AGC-controlled gain scaling factor of the adaptive postfilter
$\alpha$	The AGC factor of the adaptive postfilter
$H_H(z)$	Pre-processing high-pass filter
$w_I(n), w_{II}(n)$	LP analysis windows
$L_1^{(I)}$	Length of the first part of the LP analysis window $w_I(n)$
$L_2^{(I)}$	Length of the second part of the LP analysis window $w_I(n)$
$L_1^{(II)}$	Length of the first part of the LP analysis window $w_{II}(n)$
$L_2^{(II)}$	Length of the second part of the LP analysis window $w_{II}(n)$
$r_{ac}(k)$	The auto-correlations of the windowed speech $s'(n)$
$w_{ac}(i)$	Lag window for the auto-correlations (60 Hz bandwidth expansion)
$f_0$	The bandwidth expansion in Hz

$f_s$	The sampling frequency in Hz
$r'_{ac}(k)$	The modified (bandwidth expanded) auto-correlations
$E_{LD}(i)$	The prediction error in the $i$ th iteration of the Levinson algorithm
$k_i$	The $i$ th reflection coefficient
$\alpha_j^{(i)}$	The $j$ th direct form coefficient in the $i$ th iteration of the Levinson algorithm
$F_1'(z)$	Symmetric LSF polynomial
$F_2'(z)$	Antisymmetric LSF polynomial
$F_1(z)$	Polynomial $F_1'(z)$ with root $z = -1$ eliminated
$F_2(z)$	Polynomial $F_2'(z)$ with root $z = 1$ eliminated
$q_i$	The line spectral pairs (LSFs) in the cosine domain
$q$	An LSF vector in the cosine domain
$\hat{q}_i^{(n)}$	The quantized LSF vector at the $i$ th subframe of the frame $n$
$\omega_i$	The line spectral frequencies (LSFs)
$T_m(x)$	A $m$ th order Chebyshev polynomial
$f_1(i), f_2(i)$	The coefficients of the polynomials $F_1(z)$ and $F_2(z)$
$f_1'(i), f_2'(i)$	The coefficients of the polynomials $F_1'(z)$ and $F_2'(z)$
$f(i)$	The coefficients of either $F_1(z)$ or $F_2(z)$
$C(x)$	Sum polynomial of the Chebyshev polynomials
$x$	Cosine of angular frequency $\omega$
$\lambda_k$	Recursion coefficients for the Chebyshev polynomial evaluation
$f_i$	The line spectral frequencies (LSFs) in Hz



$\mathbf{f}' = [f_1, f_2, \dots, f_{10}]$	The vector representation of the LSFs in Hz	$\hat{s}(n)$	The gain-scaled post-filtered signal
$\mathbf{z}^{(1)}(n), \mathbf{z}^{(2)}(n)$	The mean-removed LSF vectors at frame $n$	$\hat{s}_f(n)$	Post-filtered speech signal (before scaling)
$\mathbf{r}^{(1)}(n), \mathbf{r}^{(2)}(n)$	The LSF prediction residual vectors at frame $n$	$\mathbf{x}(n)$	The target signal for adaptive codebook search
$\mathbf{p}(n)$	The predicted LSF vector at frame $n$	$\mathbf{x}_2(n), \mathbf{x}_2'$	The target signal for Fixed codebook search
$\mathbf{r}^{(2)}(n-1)$	The quantized second residual vector at the past frame	$res_{LP}(n)$	The LP residual signal
$\hat{\mathbf{r}}^k$	The quantized LSF vector at quantization index $k$	$\mathbf{c}(n)$	The fixed codebook vector
$E_{LSP}$	The LSF quantization error	$\mathbf{v}(n)$	The adaptive codebook vector
$w_i, i = 1, \dots, 10$	LSF-quantization weighting factors	$\mathbf{y}(n) = \mathbf{v}(n) * \mathbf{h}(n)$	The filtered adaptive codebook vector
$d_i$	The distance between the line spectral frequencies $f_{i+1}$ and $f_{i-1}$		The filtered fixed codebook vector
$\mathbf{h}(n)$	The impulse response of the weighted synthesis filter	$\mathbf{y}_k(n)$	The past filtered excitation
$O_k$	The correlation maximum of open-loop pitch analysis at delay $k$	$\mathbf{u}(n)$	The excitation signal
$O_i, i = 1, \dots, 3$	The correlation maxima at delays $t_i, i = 1, \dots, 3$	$\hat{\mathbf{u}}(n)$	The fully quantized excitation signal
$(M_i, t_i), i = 1, \dots, 3$	The normalized correlation maxima $M_i$ and the corresponding delays $t_i, i = 1, \dots, 3$	$\hat{\mathbf{u}}'(n)$	The gain-scaled emphasized excitation signal
$H(z)W(z) = \frac{A(z/\gamma_1)}{A(z)A(z/\gamma_2)}$	The weighted synthesis filter	$T_{op}$	The best open-loop lag
$A(z/\gamma_1)$	The numerator of the perceptual weighting filter	$t_{min}$	Minimum lag search value
$1/A(z/\gamma_2)$	The denominator of the perceptual weighting filter	$t_{max}$	Maximum lag search value
$T_1$	The nearest integer to the fractional pitch lag of the previous (1st or 3rd) subframe	$R(k)$	Correlation term to be maximized in the adaptive codebook search
$\mathbf{s}'(n)$	The windowed speech signal	$R(k)_i$	The interpolated value of $R(k)$ for the integer delay $k$ and fraction $i$
$\mathbf{s}_e(n)$	The weighted speech signal	$A_k$	Correlation term to be maximized in the algebraic codebook search at index $k$
$\hat{\mathbf{s}}(n)$	Reconstructed speech signal	$C_k$	The correlation in the numerator of $A_k$ at index $k$
		$E_{Dk}$	The energy in the denominator of $A_k$ at index $k$

$\mathbf{d} = \mathbf{H}^T \mathbf{x}_i$	The correlation between the target signal $\mathbf{x}_i(n)$ and the impulse response $h(n)$ , i.e., backward filtered target	$E_i$	The mean innovation energy
$\mathbf{H}$	The lower triangular Toeplitz convolution matrix with diagonal $h(0)$ and lower diagonals $h(1), \dots, h(39)$	$R(n)$	The prediction error of the fixed-codebook gain quantization
$\Phi = \mathbf{H}^T \mathbf{H}$	The matrix of correlations of $h(n)$	$E_Q$	The quantization error of the fixed-codebook gain quantization
$d(n)$	The elements of the vector $\mathbf{d}$	$e(n)$	The states of the synthesis filter $1/\hat{A}(z)$
$\phi(i, j)$	The elements of the symmetric matrix $\Phi$	$e_w(n)$	The perceptually weighted error of the analysis-by-synthesis search
$\mathbf{c}_i$	The innovation vector	$\eta$	The gain scaling factor for the emphasized excitation
$C$	The correlation in the numerator of $A_k$	$\delta_c$	The fixed-codebook gain
$m_i$	The position of the $i$ th pulse	$\hat{\delta}_c$	The predicted fixed-codebook gain
$\hat{v}_i$	The amplitude of the $i$ th pulse	$\delta_p$	The quantized fixed codebook gain
$N_p$	The number of pulses in the fixed codebook excitation	$\hat{\delta}_p$	The adaptive codebook gain
$E_D$	The energy in the denominator of $A_k$	$\gamma_{gc} = \delta_c / \hat{\delta}_c$	The quantized adaptive codebook gain
$res_{LTP}(n)$	The normalized long-term prediction residual	$\hat{\gamma}_{gc}$	A correction factor between the gain $\delta_c$ and the estimated one $\hat{\delta}_c$
$b(n)$	The sum of the normalized $d(n)$ vector and normalized long-term prediction residual $res_{LTP}(n)$	$\gamma_{sc}$	The optimum value for $\gamma_{gc}$
$s_b(n)$	The sign signal for the algebraic codebook search	AGC	Gain scaling factor
$\mathbf{z}', \mathbf{z}(n)$	The fixed codebook vector convolved with $h(n)$	AMR	Adaptive Gain Control
$E(n)$	The mean-removed innovation energy (in dB)	CELP	Adaptive Multi Rate
$\bar{E}$	The mean of the innovation energy	DTX	Code Excited Linear Prediction
$\bar{E}(n)$	The predicted energy	EFR	Carrier-to-Interferer ratio
$[b_1 \ b_2 \ b_3 \ b_4]$	The MA prediction coefficients	FIR	Discontinuous Transmission
$\hat{R}(k)$	The quantized prediction error at subframe $k$	FR	Enhanced Full Rate
			Finite Impulse Response
			Full Rate

HR

Half Rate

LP

Linear Prediction

LPC

Linear Predictive Coding

LSF

Line Spectral Frequency

LSF

Line Spectral Pair

LTP

Long Term Predictor (or Long Term Prediction)

MA

Moving Average

TFO

Tandem Free Operation

VAD

Voice Activity Detection

## APPENDIX B

## Bit ordering (source coding)

Bit ordering of output bits from source encoder (11 kb/s).

Bits	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19-24	Index of 4 <sup>th</sup> LSF stage
25-32	Index of 5 <sup>th</sup> LSF stage
33-37	Index of adaptive codebook gain, 1 <sup>st</sup> subframe
38-41	Index of adaptive codebook gain, 1 <sup>st</sup> subframe
42-46	Index of adaptive codebook gain, 2 <sup>nd</sup> subframe
47-50	Index of fixed codebook gain, 2 <sup>nd</sup> subframe
51-55	Index of adaptive codebook gain, 3 <sup>rd</sup> subframe
56-59	Index of fixed codebook gain, 3 <sup>rd</sup> subframe
60-64	Index of adaptive codebook gain, 4 <sup>th</sup> subframe
65-73	Index of fixed codebook gain, 4 <sup>th</sup> subframe
74-82	Index of adaptive codebook, 3 <sup>rd</sup> subframe
83-88	Index of adaptive codebook, 3 <sup>rd</sup> subframe
89-94	Index of adaptive codebook (relative), 2 <sup>nd</sup> subframe
95-96	Index of adaptive codebook (relative), 4 <sup>th</sup> subframe
97-127	Index for LSF interpolation
128-158	Index for fixed codebook, 1 <sup>st</sup> subframe
159-189	Index for fixed codebook, 2 <sup>nd</sup> subframe
190-220	Index for fixed codebook, 3 <sup>rd</sup> subframe
	Index for fixed codebook, 4 <sup>th</sup> subframe

Bit ordering of output bits from source encoder (8 kb/s).

Bits	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19-24	Index of 4 <sup>th</sup> LSF stage
25-31	Index of fixed and adaptive codebook gain, 1 <sup>st</sup> subframe
32-38	Index of fixed and adaptive codebook gain, 2 <sup>nd</sup> subframe
39-45	Index of fixed and adaptive codebook gain, 3 <sup>rd</sup> subframe
46-52	Index of fixed and adaptive codebook gain, 4 <sup>th</sup> subframe
53-60	Index of adaptive codebook, 1 <sup>st</sup> subframe
61-68	Index of adaptive codebook, 3 <sup>rd</sup> subframe
69-73	Index of adaptive codebook (relative), 2 <sup>nd</sup> subframe
74-78	Index of adaptive codebook (relative), 4 <sup>th</sup> subframe
79-80	Index for LSF interpolation
81-100	Index for fixed codebook, 1 <sup>st</sup> subframe
101-120	Index for fixed codebook, 2 <sup>nd</sup> subframe
121-140	Index for fixed codebook, 3 <sup>rd</sup> subframe
141-160	Index for fixed codebook, 4 <sup>th</sup> subframe

Bit ordering of output bits from source encoder (6.55 bbit/s).

Bit	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19-24	Index of 4 <sup>th</sup> LSF stage
25-31	Index of fixed and adaptive codebook gains, 1 <sup>st</sup> subframe
32-38	Index of fixed and adaptive codebook gains, 2 <sup>nd</sup> subframe
39-45	Index of fixed and adaptive codebook gains, 3 <sup>rd</sup> subframe
46-51	Index of fixed and adaptive codebook gains, 4 <sup>th</sup> subframe
53	Index for mode (LTP or PP)
LTP mode	
54-61	Index of adaptive codebook, 1 <sup>st</sup> subframe
62-69	Index of adaptive codebook, 2 <sup>nd</sup> subframe
70-74	Index of adaptive codebook (relative), 2 <sup>nd</sup> subframe
75-79	Index of adaptive codebook (relative), 4 <sup>th</sup> subframe
80-81	Index for LSF interpolation
82-94	Index for fixed codebook, 1 <sup>st</sup> subframe
95-107	Index for fixed codebook, 2 <sup>nd</sup> subframe
108-120	Index for fixed codebook, 3 <sup>rd</sup> subframe
121-133	Index for fixed codebook, 4 <sup>th</sup> subframe
PP mode	
	Index of pitch

Bit ordering of output bits from source encoder (5.3 bbit/s).

Bit	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19-24	Index of 4 <sup>th</sup> LSF stage
25-31	Index of fixed and adaptive codebook gains, 1 <sup>st</sup> subframe
32-38	Index of fixed and adaptive codebook gains, 2 <sup>nd</sup> subframe
39-45	Index of fixed and adaptive codebook gains, 3 <sup>rd</sup> subframe
46-52	Index of fixed and adaptive codebook gains, 4 <sup>th</sup> subframe
53-60	Index of pitch
61-74	Index for fixed codebook, 1 <sup>st</sup> subframe
75-88	Index for fixed codebook, 2 <sup>nd</sup> subframe
89-102	Index for fixed codebook, 3 <sup>rd</sup> subframe
93-116	Index for fixed codebook, 4 <sup>th</sup> subframe

Bit ordering of output bits from source encoder (4.55 bbit/s).

Bit	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19	Index of predictor
20-23	Index of fixed and adaptive codebook gains, 1 <sup>st</sup> subframe
26-31	Index of fixed and adaptive codebook gains, 2 <sup>nd</sup> subframe
32-37	Index of fixed and adaptive codebook gains, 3 <sup>rd</sup> subframe
38-43	Index of fixed and adaptive codebook gains, 4 <sup>th</sup> subframe
44-51	Index of pitch
52-61	Index for fixed codebook, 1 <sup>st</sup> subframe
62-71	Index for fixed codebook, 2 <sup>nd</sup> subframe
72-81	Index for fixed codebook, 3 <sup>rd</sup> subframe
82-91	Index for fixed codebook, 4 <sup>th</sup> subframe

## Bit ordering (channel coding)

## APPENDIX C

Ordering of bits according to subjective importance (11 bbit/s FR1CH).

Bit, see table XXX	Description
1	hfr1-0
2	hfr1-1
3	hfr1-2
4	hfr1-3
5	hfr1-4
6	hfr1-5
7	hfr2-0
8	hfr2-1
9	hfr2-2
10	hfr2-3
11	hfr2-4
12	hfr2-5
65	pitch1-0
66	pitch1-1
67	pitch1-2
68	pitch1-3
69	pitch1-4
70	pitch1-5
71	pitch3-0
72	pitch3-1
73	pitch3-2
74	pitch3-3
75	pitch3-4
76	pitch3-5
77	pitch3-6
78	pitch3-7
79	pitch3-8
80	pitch3-9
81	pitch3-10
82	pitch3-11
83	pitch3-12
84	pitch3-13
85	pitch3-14
86	pitch3-15
87	pitch3-16
88	pitch3-17

89		pitch4.0
90		pitch4.1
91		pitch4.2
92		pitch4.3
93		pitch4.4
94		pitch4.5
95		pitch4.6
96		pitch4.7
97		pitch4.8
98		pitch4.9
99		pitch5.0
100		pitch5.1
101		pitch5.2
102		pitch5.3
103		pitch5.4
104		pitch5.5
105		pitch5.6
106		pitch5.7
107		pitch5.8
108		pitch5.9
109		pitch6.0
110		pitch6.1
111		pitch6.2
112		pitch6.3
113		pitch6.4
114		pitch6.5
115		pitch6.6
116		pitch6.7
117		pitch6.8
118		pitch6.9
119		pitch7.0
120		pitch7.1
121		pitch7.2
122		pitch7.3
123		pitch7.4
124		pitch7.5
125		pitch7.6
126		pitch7.7
127		pitch7.8
128		pitch7.9
129		pitch8.0

130		exc2.2
131		exc2.3
132		exc2.4
133		exc2.5
134		exc2.6
135		exc2.7
136		exc2.8
137		exc2.9
138		exc2.10
139		exc2.11
140		exc2.12
141		exc2.13
142		exc2.14
143		exc2.15
144		exc2.16
145		exc2.17
146		exc2.18
147		exc2.19
148		exc2.20
149		exc2.21
150		exc2.22
151		exc2.23
152		exc2.24
153		exc2.25
154		exc2.26
155		exc2.27
156		exc2.28
157		exc2.29
158		exc2.30
159		exc2.31
160		exc2.32
161		exc2.33
162		exc2.34
163		exc2.35
164		exc2.36
165		exc2.37
166		exc2.38
167		exc2.39
168		exc2.40
169		exc2.41
170		exc2.42
171		exc2.43
172		exc2.44
173		exc2.45
174		exc2.46
175		exc2.47
176		exc2.48
177		exc2.49
178		exc2.50
179		exc2.51
180		exc2.52
181		exc2.53
182		exc2.54
183		exc2.55
184		exc2.56
185		exc2.57
186		exc2.58
187		exc2.59
188		exc2.60
189		exc2.61
190		exc2.62
191		exc2.63
192		exc2.64
193		exc2.65
194		exc2.66
195		exc2.67
196		exc2.68
197		exc2.69
198		exc2.70

199	etcd-9
200	etcd-10
201	etcd-11
202	etcd-12
203	etcd-13
204	etcd-14
205	etcd-15
206	etcd-16
207	etcd-17
208	etcd-18
209	etcd-19
210	etcd-20
211	etcd-21
212	etcd-22
213	etcd-23
214	etcd-24
215	etcd-25
216	etcd-26
217	etcd-27
218	etcd-28
37	fc1-4
46	fc2-4
55	fc3-4
64	fc4-4
126	etcl-29
127	etcl-30
128	etcl-31
129	etcl-32
130	etcl-33
131	etcl-34
132	etcl-35
133	etcl-36
134	etcl-37
135	etcl-38
136	etcl-39
137	etcl-40
138	etcl-41
139	etcl-42
140	etcl-43
141	etcl-44
142	etcl-45
143	etcl-46
144	etcl-47
145	etcl-48
146	etcl-49
147	etcl-50
148	etcl-51
149	etcl-52
150	etcl-53
151	etcl-54
152	etcl-55
153	etcl-56
154	etcl-57
155	etcl-58
156	etcl-59
157	etcl-60
158	etcl-61
159	etcl-62
160	etcl-63
161	etcl-64
162	etcl-65
163	etcl-66
164	etcl-67
165	etcl-68
166	etcl-69
167	etcl-70
168	etcl-71
169	etcl-72
170	etcl-73
171	etcl-74
172	etcl-75
173	etcl-76
174	etcl-77
175	etcl-78
176	etcl-79
177	etcl-80
178	etcl-81
179	etcl-82
180	etcl-83
181	etcl-84
182	etcl-85
183	etcl-86
184	etcl-87
185	etcl-88
186	etcl-89
187	etcl-90
188	etcl-91
189	etcl-92
190	etcl-93
191	etcl-94
192	etcl-95
193	etcl-96
194	etcl-97
195	etcl-98
196	etcl-99
197	etcl-100
198	etcl-101
199	etcl-102
200	etcl-103
201	etcl-104
202	etcl-105
203	etcl-106
204	etcl-107
205	etcl-108
206	etcl-109
207	etcl-110
208	etcl-111
209	etcl-112
210	etcl-113
211	etcl-114
212	etcl-115
213	etcl-116
214	etcl-117
215	etcl-118
216	etcl-119
217	etcl-120
218	etcl-121
219	etcl-122
220	etcl-123
221	etcl-124
222	etcl-125
223	etcl-126
224	etcl-127
225	etcl-128
226	etcl-129
227	etcl-130
228	etcl-131
229	etcl-132
230	etcl-133
231	etcl-134
232	etcl-135
233	etcl-136
234	etcl-137
235	etcl-138
236	etcl-139
237	etcl-140
238	etcl-141
239	etcl-142
240	etcl-143
241	etcl-144
242	etcl-145
243	etcl-146
244	etcl-147
245	etcl-148
246	etcl-149
247	etcl-150
248	etcl-151
249	etcl-152
250	etcl-153
251	etcl-154
252	etcl-155
253	etcl-156
254	etcl-157
255	etcl-158
256	etcl-159
257	etcl-160
258	etcl-161
259	etcl-162
260	etcl-163
261	etcl-164
262	etcl-165
263	etcl-166
264	etcl-167
265	etcl-168
266	etcl-169
267	etcl-170
268	etcl-171
269	etcl-172
270	etcl-173
271	etcl-174
272	etcl-175
273	etcl-176
274	etcl-177
275	etcl-178
276	etcl-179
277	etcl-180
278	etcl-181
279	etcl-182
280	etcl-183
281	etcl-184
282	etcl-185
283	etcl-186
284	etcl-187
285	etcl-188
286	etcl-189
287	etcl-190
288	etcl-191
289	etcl-192
290	etcl-193
291	etcl-194
292	etcl-195
293	etcl-196
294	etcl-197
295	etcl-198
296	etcl-199
297	etcl-200
298	etcl-201
299	etcl-202
300	etcl-203
301	etcl-204
302	etcl-205
303	etcl-206
304	etcl-207
305	etcl-208
306	etcl-209
307	etcl-210
308	etcl-211
309	etcl-212
310	etcl-213
311	etcl-214
312	etcl-215
313	etcl-216
314	etcl-217
315	etcl-218
316	etcl-219
317	etcl-220
318	etcl-221
319	etcl-222
320	etcl-223
321	etcl-224
322	etcl-225
323	etcl-226
324	etcl-227
325	etcl-228
326	etcl-229
327	etcl-230
328	etcl-231
329	etcl-232
330	etcl-233
331	etcl-234
332	etcl-235
333	etcl-236
334	etcl-237
335	etcl-238
336	etcl-239
337	etcl-240
338	etcl-241
339	etcl-242
340	etcl-243
341	etcl-244
342	etcl-245
343	etcl-246
344	etcl-247
345	etcl-248
346	etcl-249
347	etcl-250
348	etcl-251
349	etcl-252
350	etcl-253
351	etcl-254
352	etcl-255
353	etcl-256
354	etcl-257
355	etcl-258
356	etcl-259
357	etcl-260
358	etcl-261
359	etcl-262
360	etcl-263
361	etcl-264
362	etcl-265
363	etcl-266
364	etcl-267
365	etcl-268
366	etcl-269
367	etcl-270
368	etcl-271
369	etcl-272
370	etcl-273
371	etcl-274
372	etcl-275
373	etcl-276
374	etcl-277
375	etcl-278
376	etcl-279
377	etcl-280
378	etcl-281
379	etcl-282
380	etcl-283
381	etcl-284
382	etcl-285
383	etcl-286
384	etcl-287
385	etcl-288
386	etcl-289
387	etcl-290
388	etcl-291
389	etcl-292
390	etcl-293
391	etcl-294
392	etcl-295
393	etcl-296
394	etcl-297
395	etcl-298
396	etcl-299
397	etcl-300
398	etcl-301
399	etcl-302
400	etcl-303
401	etcl-304
402	etcl-305
403	etcl-306
404	etcl-307
405	etcl-308
406	etcl-309
407	etcl-310
408	etcl-311
409	etcl-312
410	etcl-313
411	etcl-314
412	etcl-315
413	etcl-316
414	etcl-317
415	etcl-318
416	etcl-319
417	etcl-320
418	etcl-321
419	etcl-322
420	etcl-323
421	etcl-324
422	etcl-325
423	etcl-326
424	etcl-327
425	etcl-328
426	etcl-329
427	etcl-330
428	etcl-331
429	etcl-332
430	etcl-333
431	etcl-334
432	etcl-335
433	etcl-336
434	etcl-337
435	etcl-338
436	etcl-339
437	etcl-340
438	etcl-341
439	etcl-342
440	etcl-343
441	etcl-344
442	etcl-345
443	etcl-346
444	etcl-347
445	etcl-348
446	etcl-349
447	etcl-350
448	etcl-351
449	etcl-352
450	etcl-353
451	etcl-354
452	etcl-355
453	etcl-356
454	etcl-357
455	etcl-358
456	etcl-359
457	etcl-360
458	etcl-361
459	etcl-362
460	etcl-363
461	etcl-364
462	etcl-365
463	etcl-366
464	etcl-367
465	etcl-368
466	etcl-369
467	etcl-370
468	etcl-371
469	etcl-372
470	etcl-373
471	etcl-374
472	etcl-375
473	etcl-376
474	etcl-377
475	etcl-378
476	etcl-379
477	etcl-380
478	etcl-381
479	etcl-382
480	etcl-383
481	etcl-384
482	etcl-385
483	etcl-386
484	etcl-387
485	etcl-388
486	etcl-389
487	etcl-390
488	etcl-391
489	etcl-392
490	etcl-393
491	etcl-394
492	etcl-395
493	etcl-396
494	etcl-397
495	etcl-398
496	etcl-399
497	etcl-400
498	etcl-401
499	etcl-402
500	etcl-403
501	etcl-404
502	etcl-405
503	etcl-406
504	etcl-407
505	etcl-408
506	etcl-409
507	etcl-410
508	etcl-411
509	etcl-412
510	etcl-413
511	etcl-414
512	etcl-415
513	etcl-416
514	etcl-417
515	etcl-418
516	etcl-419
517	etcl-420
518	etcl-421
519	etcl-422
520	etcl-423
521	etcl-424
522	etcl-425
523	etcl-426
524	etcl-427
525	etcl-428
526	etcl-429
527	etcl-430
528	etcl-431
529	etcl-432
530	etcl-433
531	etcl-434
532	etcl-435
533	etcl-436
534	etcl-437
535	etcl-438
536	etcl-439
537	etcl-440
538	etcl-441
539	etcl-442
540	etcl-443
541	etcl-444
542	etcl-445
543	etcl-446
544	etcl-447
545	etcl-448
546	etcl-449
547	etcl-450
548	etcl-451
549	etcl-452
550	etcl-453
551	etcl-454
552	etcl-455
553	etcl-456
554	etcl-457
555	etcl-458
556	etcl-459
557	etcl-460
558	etcl-461
559	etcl-462
560	etcl-463
561	etcl-464
562	etcl-465
563	etcl-466
564	etcl-467
565	etcl-468
566	etcl-469
567	etcl-470
568	etcl-471
569	etcl-472
570	etcl-473
571	etcl-474
572	etcl-475
573	etcl-476
574	etcl-477
575	etcl-478
576	etcl-479
577	etcl-480
578	etcl-481
579	etcl-482
580	etcl-483
581	etcl-484
582	etcl-485
583	etcl-486
584	etcl-487
585	etcl-488
586	etcl-489
587	etcl-490
588	etcl-491
589	etcl-492
590	etcl-493
591	etcl-494
592	etcl-495
593	etcl-496
594	etcl-497
595	etcl-498
596	etcl-499
597	etcl-500
598	etcl-501
599	etcl-502
600	etcl-503
601	etcl-504
602	etcl-505
603	etcl-506
604	etcl-507
605	etcl-508
606	etcl-509
607	etcl-510
608	etcl-511
609	etcl-512
610	etcl-513
611	etcl-514
612	etcl-515



72		pitch2-3
73		pitch4-3
74		interp-0
75		interp-1
76		interp-2
77		interp-3
78		interp-4
79		interp-5
80		interp-6
81		interp-7
82		interp-8
83		interp-9
84		interp-10
85		interp-11
86		interp-12
87		interp-13
88		interp-14
89		interp-15
90		interp-16
91		interp-17
92		interp-18
93		interp-19
94		interp-20
95		interp-21
96		interp-22
97		interp-23
98		interp-24
99		interp-25
100		interp-26
101		interp-27
102		interp-28
103		interp-29
104		interp-30
105		interp-31
106		interp-32
107		interp-33
108		interp-34
109		interp-35
110		interp-36
111		interp-37
112		interp-38
113		interp-39
114		interp-40
115		interp-41
116		interp-42
117		interp-43
118		interp-44
119		interp-45
120		interp-46
121		interp-47
122		interp-48
123		interp-49
124		interp-50
125		interp-51
126		interp-52
127		interp-53

128		exc3-7
129		exc3-8
130		exc3-9
131		exc3-10
132		exc3-11
133		exc3-12
134		exc3-13
135		exc3-14
136		exc3-15
137		exc3-16
138		exc3-17
139		exc3-18
140		exc3-19
141		exc4-0
142		exc4-1
143		exc4-2
144		exc4-3
145		exc4-4
146		exc4-5
147		exc4-6
148		exc4-7
149		exc4-8
150		exc4-9
151		exc4-10
152		exc4-11
153		exc4-12
154		exc4-13
155		exc4-14
156		exc4-15
157		exc4-16
158		exc4-17
159		exc4-18
160		exc4-19

Ordering of bits according to subjective importance (6.65 kHz, FR-TCM).

Bits, see table XXX.

Bit	Description
54	pitch-0
55	pitch-1
56	pitch-2
57	pitch-3
58	pitch-4
59	pitch-5
1	hfr-0
2	hfr-1
3	hfr-2
4	hfr-3
5	hfr-4
6	hfr-5
25	hfr1-0
26	hfr1-1
27	hfr1-2
28	hfr1-3
32	hfr2-0
33	hfr2-1
34	hfr2-2
35	hfr2-3
39	hfr3-0
40	hfr3-1
41	hfr3-2
42	hfr3-3
46	hfr4-0
47	hfr4-1
48	hfr4-2
49	hfr4-3
29	hfr1-4
36	hfr2-4
43	hfr3-4
50	hfr4-4
53	mode-0
98	exc1-0 pitch-0 (Second subframe)
99	exc1-1 pitch-1 (Second subframe)
7	hfr2-0
8	hfr2-1
9	hfr2-2
10	hfr2-3
11	hfr2-4
12	hfr2-5
30	hfr1-5
37	hfr3-5
44	hfr4-5
51	hfr4-6
62	exc1-0 pitch-0 (Third subframe)
63	exc1-1 pitch-1 (Third subframe)
64	exc1-2 pitch-2 (Third subframe)
65	exc1-3 pitch-3 (Third subframe)
66	exc1-4 pitch-4 (Third subframe)
80	exc1-0 pitch-5 (Third subframe)
100	exc1-2 pitch-2 (Second subframe)
116	exc1-0 pitch-0 (Fourth subframe)
117	exc1-1 pitch-1 (Fourth subframe)
118	exc1-2 pitch-2 (Fourth subframe)
13	hfr-0
14	hfr-1
15	hfr-2
16	hfr-3
17	hfr-4
18	hfr-5
19	hfr-6
20	hfr-7

21	hfr-2
22	hfr-3
67	exc1-5 exc1(lbp)
68	exc1-6 exc1(lbp)
69	exc1-7 exc1(lbp)
70	exc1-8 exc1(lbp)
71	exc1-9 exc1(lbp)
72	exc1-10 exc1(lbp)
81	exc2-1 exc2(lbp)
82	exc2-2 exc2(lbp)
83	exc2-3 exc2(lbp)
84	exc2-4 exc2(lbp)
85	exc2-5 exc2(lbp)
86	exc2-6 exc2(lbp)
87	exc2-7 exc2(lbp)
88	exc2-8 exc2(lbp)
89	exc2-9 exc2(lbp)
90	exc2-10 exc2(lbp)
101	exc3-3 exc3(lbp)
102	exc3-4 exc3(lbp)
103	exc3-5 exc3(lbp)
104	exc3-6 exc3(lbp)
105	exc3-7 exc3(lbp)
106	exc3-8 exc3(lbp)
107	exc3-9 exc3(lbp)
108	exc3-10 exc3(lbp)
119	exc4-3 exc4(lbp)
120	exc4-4 exc4(lbp)
121	exc4-5 exc4(lbp)
122	exc4-6 exc4(lbp)
123	exc4-7 exc4(lbp)
124	exc4-8 exc4(lbp)
125	exc4-9 exc4(lbp)
126	exc4-10 exc4(lbp)
73	exc1-11 exc1(lbp)
91	exc2-11 exc2(lbp)
109	exc3-11 exc3(lbp)
127	exc4-11 exc4(lbp)
74	exc1-12 exc1(lbp)
92	exc2-12 exc2(lbp)
110	exc3-12 exc3(lbp)
128	exc4-12 exc4(lbp)
60	pitch-6
61	pitch-7
23	hfr4-4
24	hfr4-5
75	exc1-13 exc1(lbp)
93	exc2-13 exc2(lbp)
111	exc3-13 exc3(lbp)
129	exc4-13 exc4(lbp)
31	hfr1-6
38	hfr2-6
45	hfr3-6
52	hfr4-6
76	exc1-14 exc1(lbp)
77	exc1-15 exc1(lbp)
94	exc2-14 exc2(lbp)
95	exc2-15 exc2(lbp)
112	exc3-14 exc3(lbp)
113	exc3-15 exc3(lbp)
130	exc4-14 exc4(lbp)
131	exc4-15 exc4(lbp)
78	exc1-16 exc1(lbp)
96	exc2-16 exc2(lbp)
114	exc3-16 exc3(lbp)

132	exc4-16
79	exc1-17
97	exc2-17
115	exc3-17
133	exc4-17

Ordering of bits according to subjective importance (3.8 kb/s FRITCH).

Bit, see table XXX	Description
51	pitch-0
54	pitch-1
55	pitch-2
56	pitch-3
57	pitch-4
58	pitch-5
1	lrf1-0
2	lrf1-1
3	lrf1-2
4	lrf1-3
5	lrf1-4
6	lrf1-5
7	lrf2-0
8	lrf2-1
9	lrf2-2
10	lrf2-3
11	lrf2-4
12	lrf2-5
25	rain1-0
26	rain1-1
27	rain1-2
28	rain1-3
29	rain1-4
32	rain2-0
33	rain2-1
34	rain2-2
35	rain2-3
36	rain2-4
39	rain3-0
40	rain3-1
41	rain3-2
42	rain3-3
43	rain3-4
46	rain4-0
47	rain4-1
48	rain4-2
49	rain4-3
50	rain4-4
50	rain1-5
37	rain2-5
44	rain3-5
51	rain4-5
13	lrf3-0
14	lrf3-1
15	lrf3-2
16	lrf3-3
17	lrf3-4
18	lrf3-5
59	pitch-6
60	pitch-7
19	lrf4-0
20	lrf4-1
21	lrf4-2
22	lrf4-3
23	lrf4-4
24	lrf4-5

31	rain1-6
38	rain2-6
45	rain3-6
52	rain4-6
61	exc1-0
75	exc2-0
89	exc3-0
103	exc4-0
62	exc1-1
63	exc1-2
64	exc1-3
65	exc1-4
66	exc1-5
67	exc1-6
68	exc1-7
69	exc1-8
70	exc1-9
71	exc1-10
72	exc1-11
73	exc1-12
74	exc1-13
76	exc2-1
77	exc2-2
78	exc2-3
79	exc2-4
80	exc2-5
81	exc2-6
82	exc2-7
83	exc2-8
84	exc2-9
85	exc2-10
86	exc2-11
87	exc2-12
88	exc2-13
90	exc3-1
91	exc3-2
92	exc3-3
93	exc3-4
94	exc3-5
95	exc3-6
96	exc3-7
97	exc3-8
98	exc3-9
99	exc3-10
100	exc3-11
101	exc3-12
102	exc3-13
104	exc4-1
105	exc4-2
106	exc4-3
107	exc4-4
108	exc4-5
109	exc4-6
110	exc4-7
111	exc4-8
112	exc4-9
113	exc4-10
114	exc4-11
115	exc4-12
116	exc4-13

Ordering of bits according to subjective importance (8.0 bits/HR TCH).

Bit, see table XXX	Description
1	hfr0
2	hfr1
3	hfr2
4	hfr3
5	hfr4
6	hfr5
7	hfr6
8	hfr7
9	hfr8
10	hfr9
11	hfr10
12	hfr11
13	hfr12
14	hfr13
15	hfr14
16	hfr15
17	hfr16
18	hfr17
19	hfr18
20	hfr19
21	hfr20
22	hfr21
23	hfr22

24	hfr23
25	hfr24
26	hfr25
27	hfr26
28	hfr27
29	hfr28
30	hfr29
31	hfr30
32	hfr31
33	hfr32
34	hfr33
35	hfr34
36	hfr35
37	hfr36
38	hfr37
39	hfr38
40	hfr39
41	hfr40
42	hfr41
43	hfr42
44	hfr43
45	hfr44
46	hfr45
47	hfr46
48	hfr47
49	hfr48
50	hfr49
51	hfr50
52	hfr51
53	hfr52
54	hfr53
55	hfr54
56	hfr55
57	hfr56
58	hfr57
59	hfr58
60	hfr59
61	hfr60
62	hfr61
63	hfr62
64	hfr63
65	hfr64
66	hfr65
67	hfr66
68	hfr67
69	hfr68
70	hfr69
71	hfr70
72	hfr71
73	hfr72
74	hfr73
75	hfr74
76	hfr75
77	hfr76
78	hfr77
79	hfr78
80	hfr79
81	hfr80
82	hfr81
83	hfr82
84	hfr83
85	hfr84
86	hfr85
87	hfr86
88	hfr87
89	hfr88
90	hfr89
91	hfr90
92	hfr91
93	hfr92
94	hfr93
95	hfr94
96	hfr95
97	hfr96
98	hfr97
99	hfr98
100	hfr99
101	hfr100
102	hfr101
103	hfr102
104	hfr103
105	hfr104
106	hfr105
107	hfr106
108	hfr107
109	hfr108
110	hfr109
111	hfr110
112	hfr111
113	hfr112
114	hfr113
115	hfr114
116	hfr115
117	hfr116
118	hfr117
119	hfr118
120	hfr119
121	hfr120
122	hfr121
123	hfr122
124	hfr123
125	hfr124
126	hfr125
127	hfr126
128	hfr127

129	exc3-8
130	exc3-9
131	exc3-10
132	exc3-11
133	exc3-12
134	exc3-13
135	exc3-14
136	exc3-15
137	exc3-16
138	exc3-17
139	exc3-18
140	exc3-19
141	exc4-0
142	exc4-1
143	exc4-2
144	exc4-3
145	exc4-4
146	exc4-5
147	exc4-6
148	exc4-7
149	exc4-8
150	exc4-9
151	exc4-10
152	exc4-11
153	exc4-12
154	exc4-13
155	exc4-14
156	exc4-15
157	exc4-16
158	exc4-17
159	exc4-18
160	exc4-19

Ordering of bits according to subjective importance (6.65 bits/HRTCH).

Bits, see table XXX	Description
53	mode-0
54	pitch-0
55	pitch-1
56	pitch-2
57	pitch-3
58	pitch-4
59	pitch-5
1	brf1-0
2	brf1-1
3	brf1-2
4	brf1-3
5	brf1-4
6	brf1-5
7	brf2-0
8	brf2-1
9	brf2-2
10	brf2-3
11	brf2-4
12	brf2-5
25	gain1-0
26	gain1-1
27	gain1-2
28	gain1-3
32	gain2-0
33	gain2-1
34	gain2-2
35	gain2-3
39	gain3-0
40	gain3-1
41	gain3-2
42	gain3-3
46	gain4-0
47	gain4-1
48	gain4-2
49	gain4-3
59	gain1-4
36	gain2-4
43	gain3-4
50	gain4-4
62	exc1-0 pitch-0(Third subframe)
63	exc1-1 pitch-1(Third subframe)
64	exc1-2 pitch-2(Third subframe)
65	exc1-3 pitch-3(Third subframe)
80	exc2-0 pitch-0(Second subframe)
98	exc3-1 pitch-1(Second subframe)
99	exc3-2 pitch-2(Second subframe)
100	exc3-3 pitch-3(Second subframe)
116	exc4-1 pitch-1(Fourth subframe)
117	exc4-2 pitch-2(Fourth subframe)
118	exc4-3 pitch-3(Fourth subframe)
13	brf3-0
14	brf3-1
15	brf3-2
16	brf3-3
17	brf3-4
18	brf3-5
19	brf4-0
20	brf4-1
21	brf4-2
22	brf4-3
23	brf4-4
24	brf4-5
81	exc2-1 exc2(inp)

82	exc2.2 exc2(lbp)
83	exc2.3 exc2(lbp)
101	exc3.3 exc3(lbp)
119	exc4.3 exc4(lbp)
66	exc1.4 pitch-4(Third subframe)
102	exc3.4 exc3(lbp)
120	exc4.4 exc4(lbp)
67	exc1.5 exc1(lbp)
68	exc1.6 exc1(lbp)
69	exc1.7 exc1(lbp)
70	exc1.8 exc1(lbp)
71	exc1.9 exc1(lbp)
72	exc1.10
73	exc1.11
84	exc2.5 exc2(lbp)
86	exc2.6 exc2(lbp)
87	exc2.7
88	exc2.8
89	exc2.9
90	exc2.10
91	exc2.11
103	exc3.5 exc3(lbp)
104	exc3.6 exc3(lbp)
105	exc3.7 exc3(lbp)
106	exc3.8
107	exc3.9
108	exc3.10
109	exc3.11
121	exc4.5 exc4(lbp)
122	exc4.6 exc4(lbp)
123	exc4.7 exc4(lbp)
124	exc4.8
125	exc4.9
126	exc4.10
127	exc4.11
30	run1.5
31	run1.6
37	run2.5
38	run2.6
44	run3.5
45	run3.6
51	run4.5
52	run4.6
60	pitch-6
61	pitch-7
74	exc1.12
75	exc1.13
76	exc1.14
77	exc1.15
78	exc1.16
92	exc2.12
93	exc2.13
94	exc2.14
95	exc2.15
110	exc3.12
111	exc3.13
112	exc3.14
113	exc3.15
128	exc4.12
129	exc4.13
130	exc4.14
131	exc4.15
78	exc1.16
96	exc2.16
114	exc3.16

132	exc4.16
79	exc1.17
97	exc2.17
115	exc3.17
133	exc4.17

Ordering of bits according to subjective importance (5.8 bits HRTF).

Bits are table XXX

Bits	Description
25	run1.0
26	run1.1
32	run2.0
33	run2.1
39	run3.0
40	run3.1
46	run4.0
47	run4.1
1	br1.0
2	br1.1
3	br1.2
4	br1.3
5	br1.4
6	br1.5
37	run1.2
38	run1.3
41	run2.2
48	run2.3
53	pitch-0
54	pitch-1
55	pitch-2
56	pitch-3
57	pitch-4
58	pitch-5
28	run1.3
29	run1.4
35	run2.3
36	run2.4
42	run3.3
43	run3.4
49	run4.3
50	run4.4
7	br2.0
8	br2.1
9	br2.2
10	br2.3
11	br2.4
12	br2.5
13	br3.0
14	br3.1
15	br3.2
16	br3.3
17	br3.4
18	br3.5
19	br4.0
20	br4.1
21	br4.2
22	br4.3
30	run1.5
37	run2.5
44	run3.5
51	run4.5
31	run1.6
38	run2.6
45	run3.6
52	run4.6
61	exc1.0

63	exc1-1
63	exc1-2
64	exc1-3
75	exc2-0
76	exc2-1
77	exc2-2
78	exc2-3
89	exc3-0
90	exc3-1
91	exc3-2
92	exc3-3
103	exc4-0
104	exc4-1
105	exc4-2
106	exc4-3
23	lrf4-4
24	lrf4-5
39	pitch-6
60	pitch-7
65	exc1-4
66	exc1-5
67	exc1-6
68	exc1-7
69	exc1-8
70	exc1-9
71	exc1-10
72	exc1-11
73	exc1-12
74	exc1-13
79	exc2-4
80	exc2-5
81	exc2-6
82	exc2-7
83	exc2-8
84	exc2-9
85	exc2-10
86	exc2-11
87	exc2-12
88	exc2-13
93	exc3-4
94	exc3-5
95	exc3-6
96	exc3-7
97	exc3-8
98	exc3-9
99	exc3-10
100	exc3-11
101	exc3-12
102	exc3-13
107	exc4-4
108	exc4-5
109	exc4-6
110	exc4-7
111	exc4-8
112	exc4-9
113	exc4-10
114	exc4-11
115	exc4-12
116	exc4-13

Ordering of this according to subjective importance (4.55 kb/s HRTCH).  
Bns. see table XXX

20	gain1-0
26	gain2-0
44	pitch-0
45	pitch-1
46	pitch-2
32	gain3-0
38	gain4-0
21	gain1-1
27	gain2-1
33	gain3-1
39	gain4-1
19	exc1-lf
1	lrf1-0
2	lrf1-1
3	lrf1-2
4	lrf1-3
5	lrf1-4
6	lrf1-5
7	lrf2-0
8	lrf2-1
9	lrf2-2
22	gain1-2
28	gain2-2
34	gain3-2
40	gain4-2
23	gain1-3
29	gain2-3
35	gain3-3
41	gain4-3
47	pitch-3
10	lrf2-3
11	lrf2-4
12	lrf2-5
24	gain1-4
30	gain2-4
36	gain3-4
42	gain4-4
48	pitch-4
49	pitch-5
13	lrf3-0
14	lrf3-1
15	lrf3-2
16	lrf3-3
17	lrf3-4
18	lrf3-5
25	gain1-5
31	gain2-5
37	gain3-5
43	gain4-5
50	pitch-6
51	pitch-7
52	exc1-0
53	exc1-1
54	exc1-2
55	exc1-3
56	exc1-4
57	exc1-5
58	exc1-6
62	exc2-0
63	exc2-1
64	exc2-2
65	exc2-3
66	exc2-4

CLAIMS

I claim:

1. A speech codec using long term preprocessing of a speech signal having a pitch lag, the speech codec comprising:

an adaptive codebook;

an encoder, coupled to the adaptive codebook, that estimates the pitch lag; and

the encoder applying continuous warping of the speech signal using the estimated

pitch lag.

2. The speech codec of claim 1 wherein the speech signal comprises a weighted speech signal.

3. The speech codec of any of claims 1 and 2 wherein the encoder searches for a best local delay using linear time weighting.

4. The speech codec of any of claims 1 and 2 wherein the continuous warping comprises translating the speech signal from a first time region to a second time region.

5. The speech codec of claim 1 wherein the speech signal comprises a residual signal.

6. A speech codec using long term preprocessing of a speech signal, the speech codec comprising:  
an adaptive codebook;

67	enc2-5
72	enc1-0
73	enc1-1
74	enc1-2
75	enc1-3
76	enc1-4
77	enc1-5
82	enc4-0
83	enc4-1
84	enc4-2
85	enc4-3
86	enc4-4
87	enc4-5
89	enc1-7
90	enc1-8
91	enc1-9
68	enc2-6
69	enc2-7
70	enc2-8
71	enc2-9
78	enc3-6
79	enc3-7
80	enc3-8
81	enc3-9
88	enc4-6
89	enc4-7
90	enc4-8
91	enc4-9



an encoder, coupled to the adaptive codebook, that continuously warps the speech signal to a target contour; and  
the encoder searches for a best local delay using linear time weighting.

7. The speech codec of claim 6 wherein the speech signal comprises a weighted speech signal.
8. The speech codec of claim 6 wherein the speech signal comprises a residual signal.
9. The speech codec of claim 6 wherein the encoder processing circuit identifies a limited search range for the best local delay.
10. The speech codec of claim 9 wherein the identification by the encoder of the limited search range is based at least in part on sharpness of the speech signal.
11. The speech codec of claim 9 wherein the identification by the encoder of the limited search range is based at least in part on a classification of the speech signal.
12. The speech codec of claim 11 wherein the classification of the speech signal involves classifying the speech signal as either voiced or unvoiced speech.
13. The speech codec of claim 6 wherein the speech signal having a previous pitch lag and a current pitch lag, and the encoder utilizes estimates of the previous pitch lag and the current pitch lag to generate the target contour.

1/11

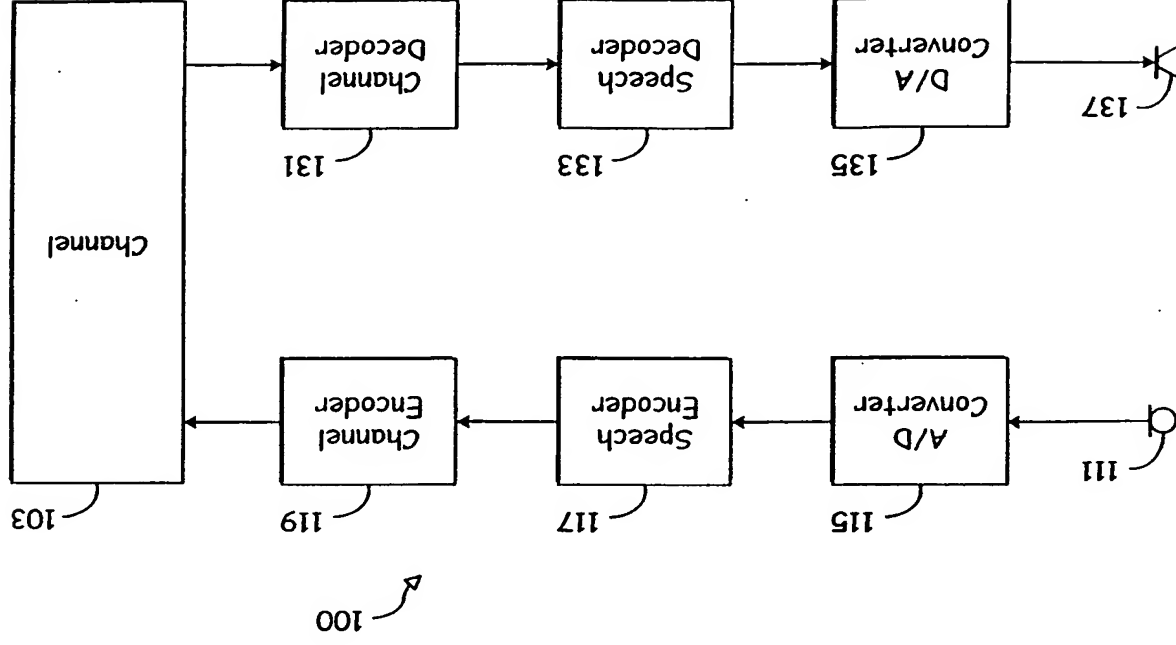


Fig. 1a

3/11

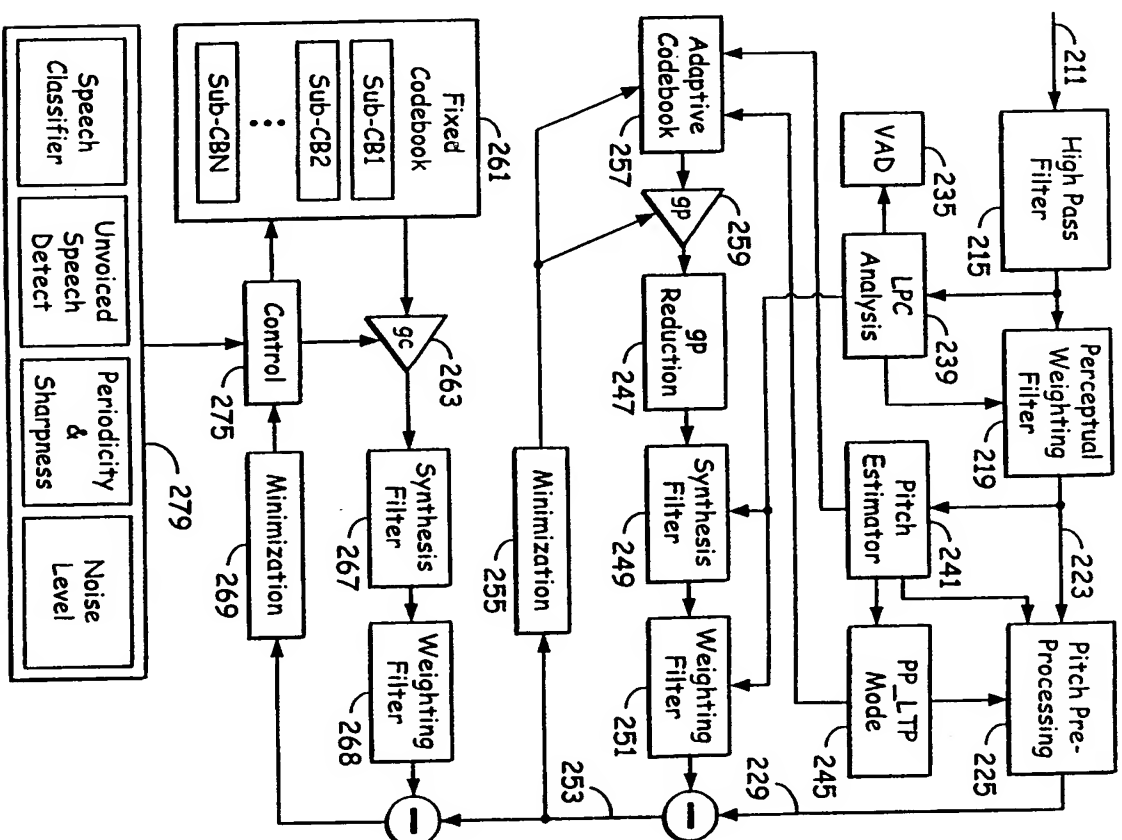


Fig. 2

5/11

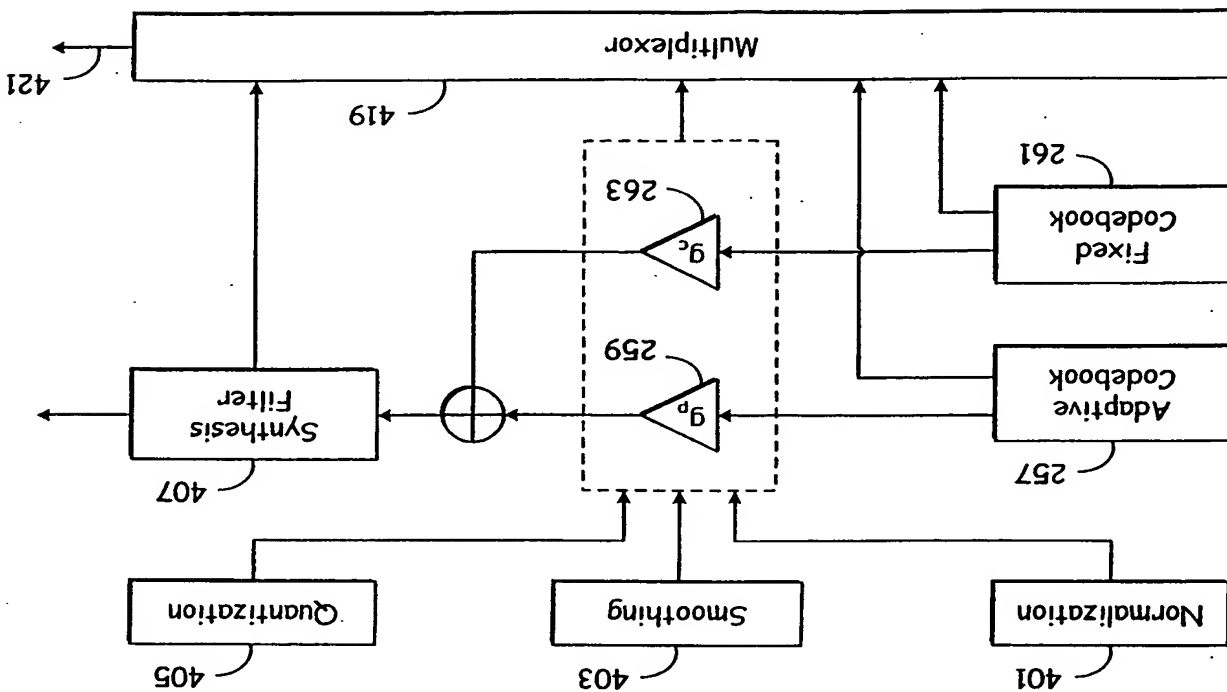
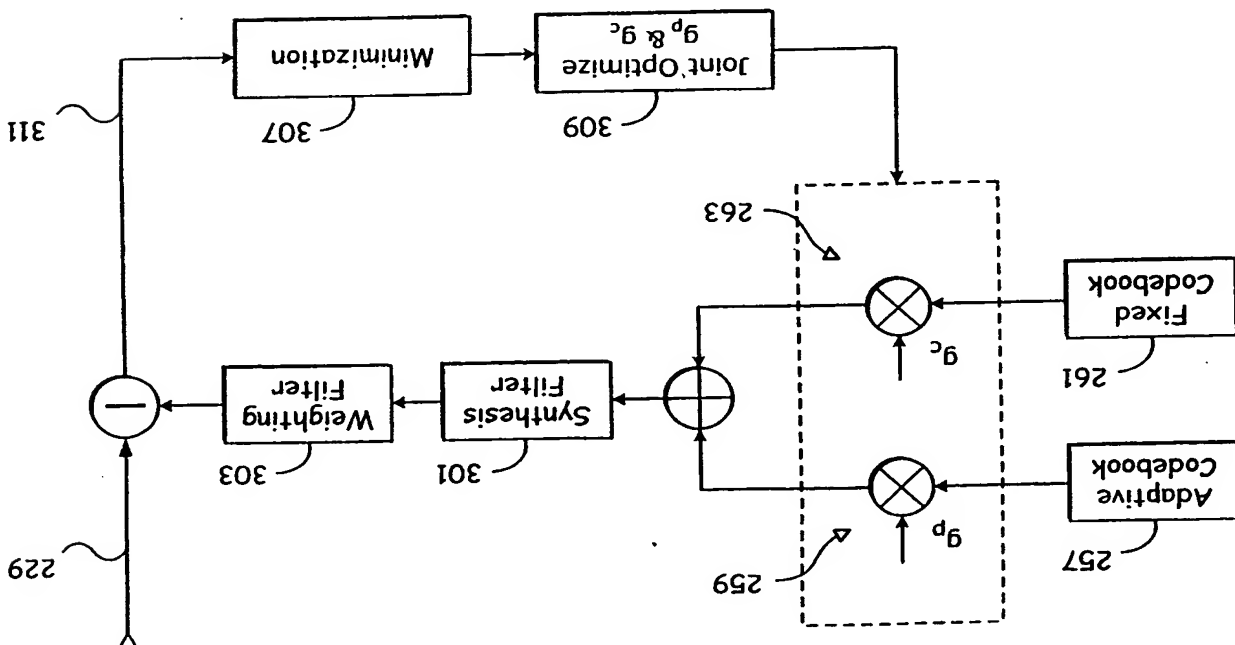


Fig. 3

4/11



6/11

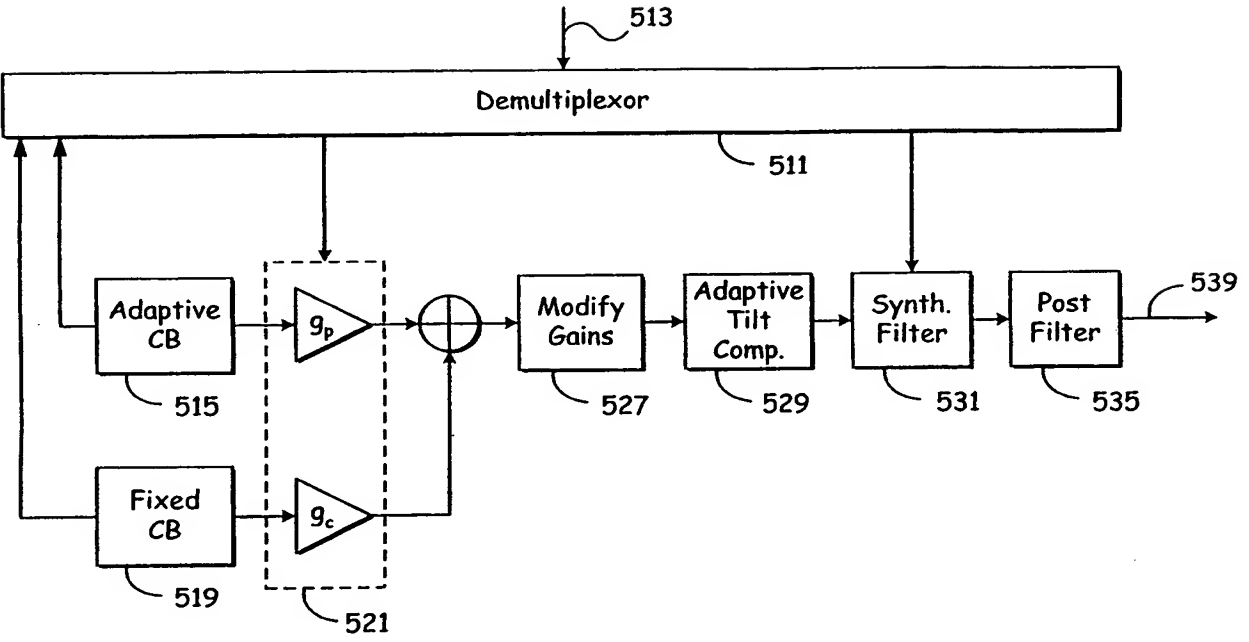


Fig. 5

7/11

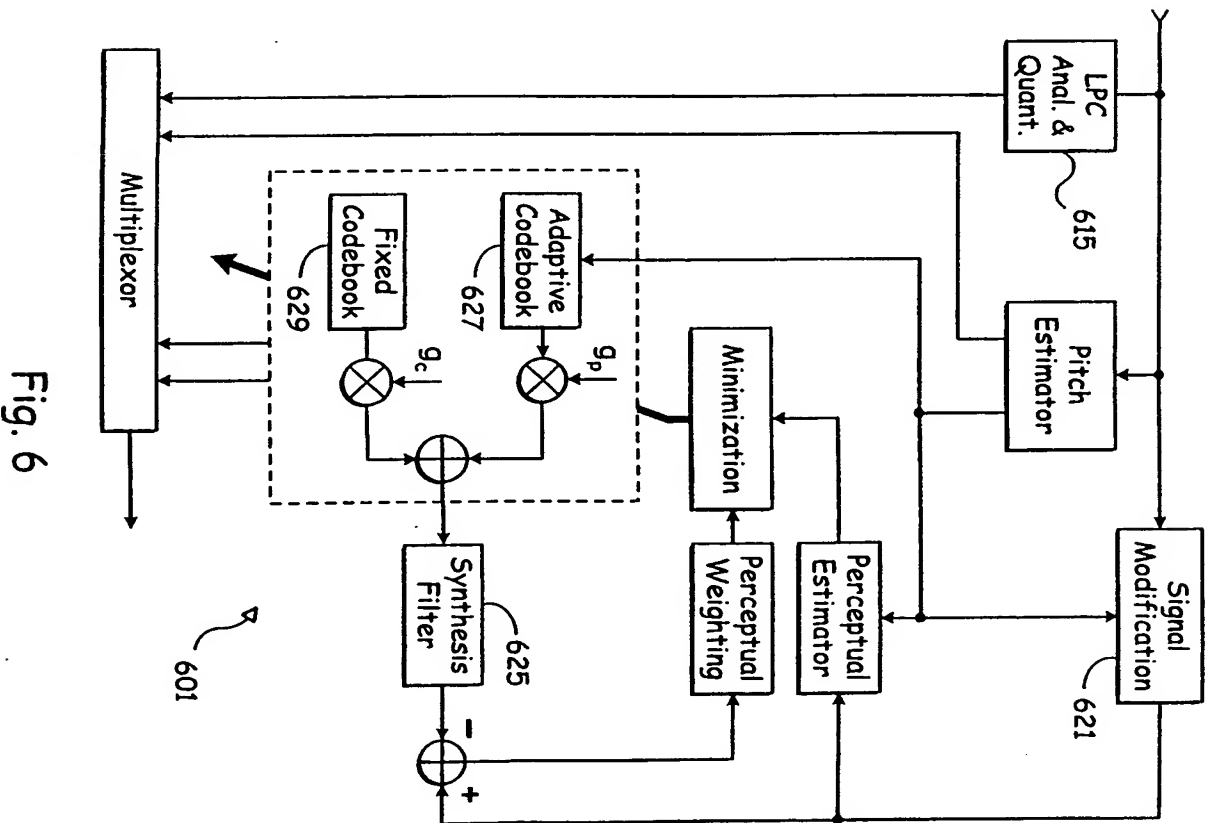


Fig. 6

9/11

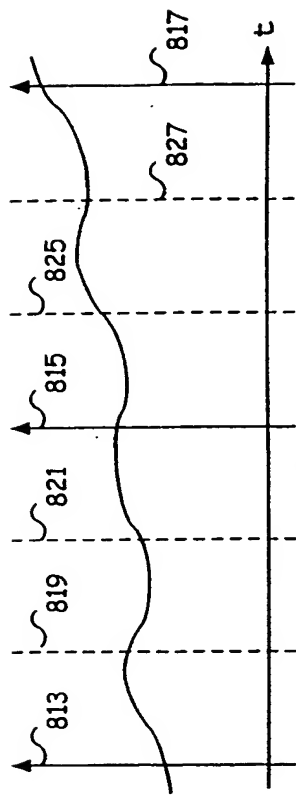


Fig. 8a

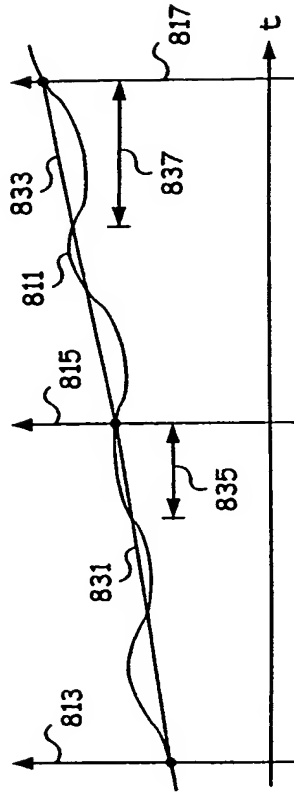


Fig. 8b

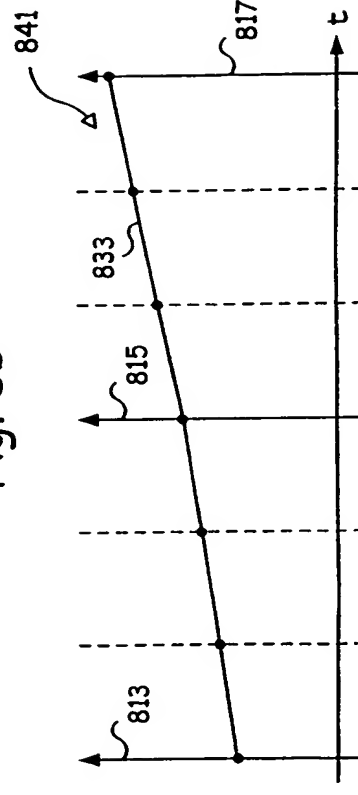
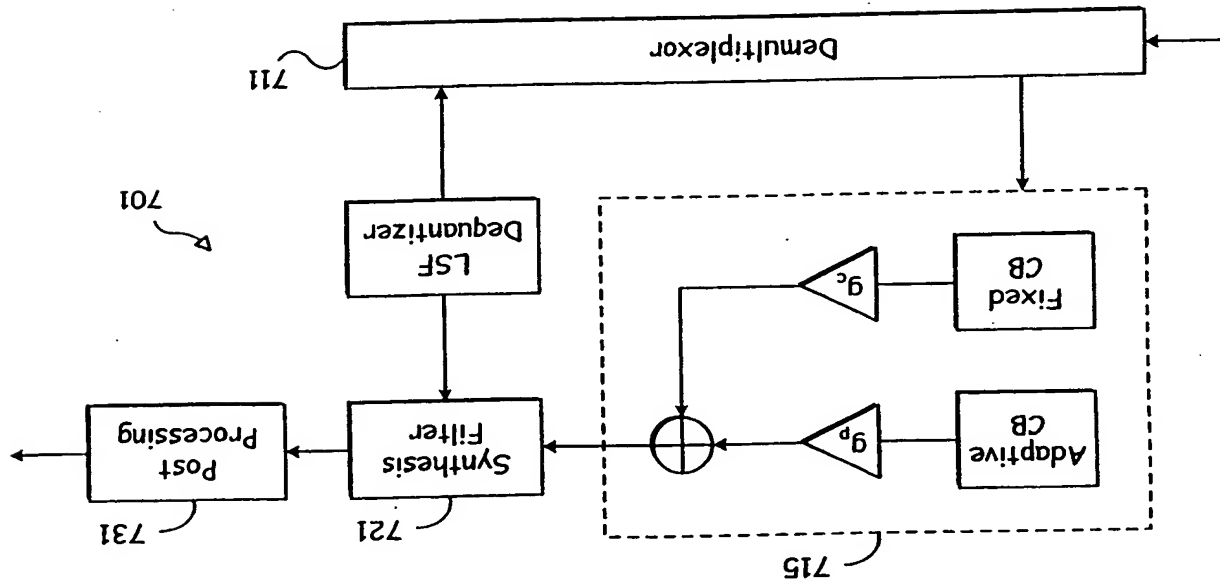


Fig. 8c

Fig. 7



8/11

10/11

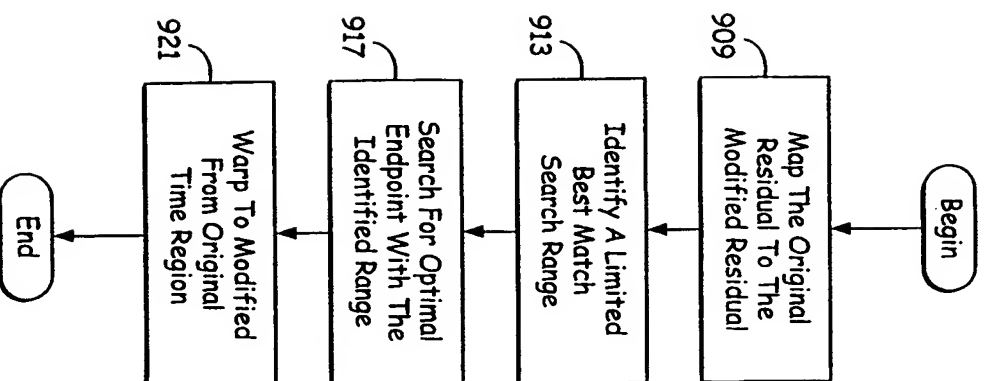


Fig. 9

11/11

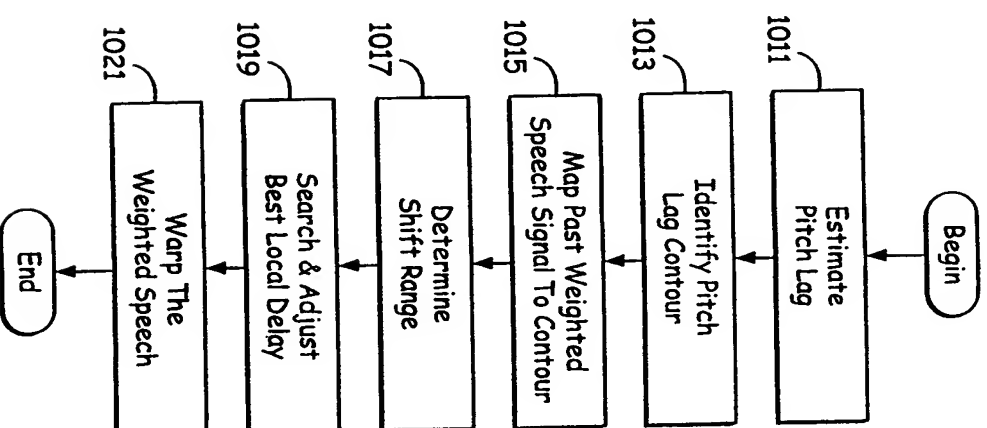


Fig. 10

# INTERNATIONAL SEARCH REPORT

International Application No.  
PCT/US 99/19175

A. CLASSIFICATION OF SUBJECT MATTER  
IPC 7 G10L19/08 G10L19/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELD OF SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
IPC 7 G10L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	KLEIJN W B ET AL: "INTERPOLATION OF THE PITCH-PREDICTOR PARAMETERS IN ANALYSIS-BY-SYNTHESIS SPEECH CODERS" IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, US, IEEE INC., NEW YORK, vol. 2, no. 1, PART I, page 42-54 XP000423486 ISSN: 1063-6676 page 46 -page 48	1-9, 13
A	ROUAT J ET AL: "A pitch determination and voiced/unvoiced decision algorithm for noisy speech" SPEECH COMMUNICATION, NL, ELSEVIER SCIENCE PUBLISHERS, AMSTERDAM, vol. 21, no. 3, page 191-207 XP004059542 ISSN: 0167-6393 page 194	1, 6, 10

☐ Further documents are listed in the continuation of box C.

☐ Patent family members are listed in annex.

### \* Special categories of cited documents:

- \*1\* document defining the general state of the art which is not considered to be of particular relevance
- \*2\* earlier document published on or after the international filing date
- \*3\* document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (see specification)
- \*4\* document referring to an oral disclosure, use, exhibition or other means
- \*5\* document published prior to the international filing date but later than the priority date claimed
- \*6\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- \*7\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- \*8\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents or such combination being obvious to a person skilled in the art
- \*9\* document member of the same patent family

Date of the actual completion of the international search

Date of mailing of the international search report

10 December 1999

11/01/2000

Name and mailing address of the ISA

European Patent Office, P.O. Box 1618, 6500 AA Eindhoven, NL  
Tel. (+31-70) 340-2040, Fax (+31-70) 340-2040, Telex (+31-70) 340-2040

Authorized officer

Ramos Sánchez, U



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification: **G10L 19/08, 19/12** (11) International Publication Number: **WO 00/11653**  
(43) International Publication Date: **2 March 2000 (02.03.00)**

(21) International Application Number: **PCT/US99/19175** (81) Designated States: **CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).**  
(22) International Filing Date: **24 August 1999 (24.08.99)**

(30) Priority Data: **60/097,569 24 August 1998 (24.08.98) US**  
**09/154,675 18 September 1998 (18.09.98) US**

**Published**  
*With international search report.  
Before the expiration of the time limit for amending the  
claims and to be republished in the event of the receipt of  
amendments.*

(71) Applicant: **CONEXANT SYSTEMS, INC. [USUS: 4311]**  
**Jamboree Road, Newport Beach, CA 92660-3095 (US).**

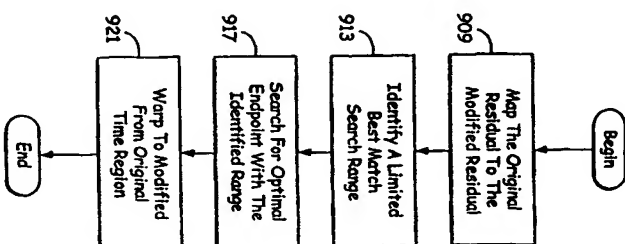
(72) Inventor: **GAO, Yang; 26586 San Torini Road, Mission Viejo,  
CA 92692-6101 (US).**

(74) Agent: **BENNETT, James, D.; Akin, Gump, Strauss, Hauer &  
Feld, L.L.P., Suite 1900, 816 Congress Avenue, Austin, TX  
78701 (US).**

(54) Title: **SPEECH ENCODER USING CONTINUOUS WARPING COMBINED WITH LONG TERM PREDICTION**

(57) Abstract

A multi-rate speech code supports a plurality of encoding bit rate modes by adaptively selecting encoding bit rate modes to match communication channel restrictions. In higher bit rate encoding modes, an accurate representation of speech through CELP (code excited linear prediction) and other associated modeling parameters are generated for higher quality decoding and reproduction. To support lower bit rate encoding modes, a variety of techniques are applied many of which involve the classification of the input signal. The speech encoder continuously warps a weighted speech signal in long term preprocessing. The continuous warping is applied to a linear pitch lag contour that enables fast searching through linear time weighting. Optimal searching is performed within a limited range that is defined at least in part on sharpness and speech classification. The speech encoder generates the linear pitch lag contour from previous and current pitch lag values. Such continuous warping may also be applied in an open loop approach to the residual signal.



Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

FOR THE PURPOSES OF INFORMATION ONLY

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AN	Antigua	FR	France	LU	Luxembourg	SN	Senegal
AT	Austria	GB	Great Britain	LV	Latvia	SZ	Swaziland
AU	Australia	GE	Georgia	MC	Monaco	TD	Togo
AZ	Azerbaijan	GR	Greece	MD	Moldova	TG	Togo
BA	Bosnia and Herzegovina	GN	Guinea	MG	Madagascar	TJ	Tajikistan
BB	Barbados	HR	Croatia	MK	Macedonia	TM	Turkmenistan
BE	Belgium	HU	Hungary	ML	Mali	TR	Turkey
BF	Burkina Faso	IE	Ireland	MN	Mongolia	TT	Trinidad and Tobago
BG	Bulgaria	IL	Israel	MR	Mauritania	UA	Ukraine
BJ	Benin	IS	Iceland	MT	Malta	UG	Uganda
BR	Brazil	IT	Italy	MW	Malawi	US	United States of America
BS	Bahamas	JP	Japan	MX	Mexico	UZ	Uzbekistan
BT	Bhutan	KE	Kenya	NE	Niger	VN	Viet Nam
BV	Bermuda	KG	Kyrgyzstan	NL	Netherlands	YU	Yugoslavia
BY	Belarus	KH	Kampuchea	NO	Norway	ZW	Zimbabwe
CA	Canada	KR	Republic of Korea	NZ	New Zealand		
CC	Cocos (Keeling) Islands	KZ	Kazakhstan	PL	Poland		
CD	Congo	LA	Laos	PT	Portugal		
CE	Cote d'Ivoire	LC	Liechtenstein	RO	Romania		
CF	Cote d'Ivoire	LI	Liechtenstein	RU	Russian Federation		
CG	Congo	LK	Sri Lanka	SE	Sweden		
CH	Switzerland	LR	Liberia	SG	Singapore		
CI	Cote d'Ivoire						
CN	China						
CU	Cuba						
CZ	Czech Republic						
DE	Germany						
DK	Denmark						
EE	Estonia						



**TITLE:**

SPEECHCODER USING CONTINUOUS WARPING COMBINED WITH LONG TERM PREDICTION

**SPECIFICATION****CROSS-REFERENCE TO RELATED APPLICATIONS**

The present application is based on U.S. Patent Application Ser. No. 09/154,675, filed September 18, 1998. This application is based on U.S. Provisional Application Serial No. 60/097,569, filed on August 24, 1998. All of such applications are hereby incorporated herein by reference in their entirety and made part of the present application.

**INCORPORATION BY REFERENCE**

The following applications are hereby incorporated herein by reference in their entirety and made part of the present application:

- 1) U.S. Provisional Application Serial No. 60/097,569 (Attorney Docket No. 98RSS325), filed August 24, 1998;
- 2) U.S. Patent Application Serial No. 09/154,675 (Attorney Docket No. 97RSS383), filed September 18, 1998;
- 3) U.S. Patent Application Serial No. 09/156,814 (Attorney Docket No. 98RSS365), filed September 18, 1998;
- 4) U.S. Patent Application Serial No. 09/156,649 (Attorney Docket No. 95E020), filed September 18, 1998;
- 5) U.S. Patent Application Serial No. 09/156,648 (Attorney Docket No. 98RSS228), filed September 18, 1998;
- 6) U.S. Patent Application Serial No. 09/156,650 (Attorney Docket No. 98RSS343), filed September 18, 1998;
- 7) U.S. Patent Application Serial No. 09/156,832 (Attorney Docket No. 97RSS039), filed September 18, 1998;

- 8) U.S. Patent Application Serial No. 09/154,654 (Attorney Docket No. 98RSS344), filed September 18, 1998;
- 9) U.S. Patent Application Serial No. 09/154,657 (Attorney Docket No. 98RSS328), filed September 18, 1998;
- 10) U.S. Patent Application Serial No. 09/156,826 (Attorney Docket No. 98RSS382), filed September 18, 1998;
- 11) U.S. Patent Application Serial No. 09/154,662 (Attorney Docket No. 98RSS383), filed September 18, 1998;
- 12) U.S. Patent Application Serial No. 09/154,653 (Attorney Docket No. 98RSS406), filed September 18, 1998;
- 13) U.S. Patent Application Serial No. 09/154,660 (Attorney Docket No. 98RSS384), filed September 18, 1998.
- 14) U.S. Patent Application Serial No. 09/198,414 (Attorney Docket No. 97RSS039CIP), filed November 24, 1998.

## BACKGROUND

### 1. Technical Field

The present invention relates generally to speech encoding and decoding in voice communication systems; and, more particularly, it relates to various techniques used with code-excited linear prediction coding to obtain high quality speech reproduction through a limited bit rate communication channel.

### 2. Related Art

Signal modeling and parameter estimation play significant roles in communicating voice information with limited bandwidth constraints. To model basic speech sounds, speech signals are sampled as a discrete waveform to be digitally processed. In one type of signal coding technique called LPC (linear predictive coding), the signal value at any particular time index is modeled as a linear function of previous values. A subsequent signal is thus linearly predictable according to an earlier value. As a result, efficient signal representations can be determined by estimating and applying certain prediction parameters to represent the signal.

Applying LPC techniques, a conventional source encoder operates on speech signals to extract modeling and parameter information for communication to a conventional source decoder via a communication channel. Once received, the decoder attempts to reconstruct a counterpart signal for playback that sounds to a human ear like the original speech.

A certain amount of communication channel bandwidth is required to communicate the modeling and parameter information to the decoder. In embodiments, for example where the channel bandwidth is shared and real-time reconstruction is necessary, a reduction in the required bandwidth proves beneficial. However, using conventional modeling techniques, the quality

requirements in the reproduced speech limit the reduction of such bandwidth below certain levels.

In conventional coding systems employing long term preprocessing, a modified residual is produced as a new reference for current excitation. The goal is to produce a modified residual that better matches a coded pitch contour (or delay contour) than the original residual so that the LTP gain is higher. This is attempted in conventional systems by individually shifting the pitch pulses to match the pitch contour, requiring reliable endpoint detection of a segment to be shifted to maintain signal continuity. Using such an open loop approach with pulse shifting results in quality problems in speech reproduction.

Additionally, in using such and other conventional approaches, the amount of pitch lag information that must be transmitted is relatively large in view of the limitations often placed on the channel bit rate. For example, 8 bits might be required to encode pitch lag for a first subframe (of 5ms duration) followed perhaps by 5 bits for pitch lag changes in a second subframe, resulting in a relatively large amount of bandwidth allocation, e.g., 1.3 kbps (kilobits per second), just for the pitch lag information.

Further limitations and disadvantages of conventional systems will become apparent to one of skill in the art after reviewing the remainder of the present application with reference to the drawings.

### SUMMARY OF THE INVENTION

Various aspects of the present invention can be found in an embodiment of a speech encoder that uses long term preprocessing of a speech signal wherein the speech signal has a previous pitch lag and a current pitch lag. Therein, the speech encoder comprises an adaptive codebook and an encoder processing circuit coupled to the adaptive codebook. Using estimates of the previous pitch lag and the current pitch lag, the encoder processing circuit generates a pitch lag contour. The encoder processing circuit continuously warps the speech signal to the pitch lag contour.

Many possible variations and further aspects of such a speech encoder are possible. For example, the speech signal may comprise either a weighted speech signal or a residual signal. The pitch lag contour may comprise a linear segment bounded by the estimates of the previous pitch lag and the current pitch lag, and continuous warping may involve warping the speech signal from a first time region to a second time region. Additionally, for example, the encoder processing circuit may search for a best local delay using linear time weighting, and/or perform the estimation of the current pitch lag.

Further aspects of the present invention may be found in an alternate embodiment of a speech encoder that uses long term preprocessing of a speech signal having a pitch lag. As before, the speech encoder comprises an adaptive codebook and an encoder processing circuit coupled thereto. The encoder processing circuit estimates the pitch lag, and, based on such estimate, applies continuous warping of the speech signal.

Other variations and further aspects such as those mentioned previously also apply to this embodiment. For example, the speech signal might comprise a weighted speech signal or a residual signal. The encoder processing circuit may search for a best local delay using linear

time weighting, or conduct continuous warping by translating the speech signal from a first time region to a second time region.

Other aspects, advantages and novel features of the present invention will become apparent from the following detailed description of the invention when considered in conjunction with the accompanying drawings.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

Fig. 1a is a schematic block diagram of a speech communication system illustrating the use of source encoding and decoding in accordance with the present invention.

Fig. 1b is a schematic block diagram illustrating an exemplary communication device utilizing the source encoding and decoding functionality of Fig. 1a.

Figs. 2-4 are functional block diagrams illustrating a multi-step encoding approach used by one embodiment of the speech encoder illustrated in Figs. 1a and 1b. In particular, Fig. 2 is a functional block diagram illustrating of a first stage of operations performed by one embodiment of the speech encoder of Figs. 1a and 1b. Fig. 3 is a functional block diagram of a second stage of operations, while Fig. 4 illustrates a third stage.

Fig. 5 is a block diagram of one embodiment of the speech decoder shown in Figs. 1a and 1b having corresponding functionality to that illustrated in Figs. 2-4.

Fig. 6 is a block diagram of an alternate embodiment of a speech encoder that is built in accordance with the present invention.

Fig. 7 is a block diagram of an embodiment of a speech decoder having corresponding functionality to that of the speech encoder of Fig. 6.

Fig. 8a is a timing diagram of an exemplary pitch lag contour over two speech frames to which continuous warping techniques are applied in accordance with the present invention.

Fig. 8b is a timing diagram illustrating a linear pitch contour to which continuous warping of the original pitch lag contour is applied in accordance with the present invention.

Fig. 8c is a timing diagram illustrating the use of the linear pitch lag contour of Fig. 8b which can be represented by a lesser number of bits than the original pitch lag contour of Fig. 8a.

Fig. 9 is a flow diagram illustrating an embodiment of the continuous warping approach and an associated fast searching process used by an encoder of the present invention to carry out the functionality described in reference to Figs. 8a-c on a residual signal using an open loop approach.

Fig. 10 is a flow diagram illustrating an alternate embodiment of functionality of a speech encoder of the present invention that performs continuous warping to the weighted speech signal in a closed loop approach.

# DETAILED DESCRIPTION

Fig. 1a is a schematic block diagram of a speech communication system illustrating the use of source encoding and decoding in accordance with the present invention. Therein, a speech communication system 100 supports communication and reproduction of speech across a communication channel 103. Although it may comprise for example a wire, fiber or optical link, the communication channel 103 typically comprises, at least in part, a radio frequency link that often must support multiple, simultaneous speech exchanges requiring shared bandwidth resources such as may be found with cellular telephony embodiments.

Although not shown, a storage device may be coupled to the communication channel 103 to temporarily store speech information for delayed reproduction or playback, e.g., to perform answering machine functionality, voicemail, etc. Likewise, the communication channel 103 might be replaced by such a storage device in a single device embodiment of the communication system 100 that, for example, merely records and stores speech for subsequent playback.

In particular, a microphone 111 produces a speech signal in real time. The microphone 111 delivers the speech signal to an A/D (analog to digital) converter 115. The A/D converter 115 converts the speech signal to a digital form then delivers the digitized speech signal to a speech encoder 117.

The speech encoder 117 encodes the digitized speech by using a selected one of a plurality of encoding modes. Each of the plurality of encoding modes utilizes particular techniques that attempt to optimize quality of resultant reproduced speech. While operating in any of the plurality of modes, the speech encoder 117 produces a series of modeling and parameter information (hereinafter "speech indices"), and delivers the speech indices to a channel encoder 119.

-9-

The channel encoder 119 coordinates with a channel decoder 131 to deliver the speech indices across the communication channel 103. The channel decoder 131 forwards the speech indices to a speech decoder 133. While operating in a mode that corresponds to that of the speech encoder 117, the speech decoder 133 attempts to recreate the original speech from the speech indices as accurately as possible at a speaker 137 via a D/A (digital to analog) converter 135.

The speech encoder 117 adaptively selects one of the plurality of operating modes based on the data rate restrictions through the communication channel 103. The communication channel 103 comprises a bandwidth allocation between the channel encoder 119 and the channel decoder 131. The allocation is established, for example, by telephone switching networks wherein many such channels are allocated and reallocated as need arises. In one such embodiment, either a 22.8 kbps (kilobits per second) channel bandwidth, i.e., a full rate channel, or a 11.4 kbps channel bandwidth, i.e., a half rate channel, may be allocated.

With the full rate channel bandwidth allocation, the speech encoder 117 may adaptively select an encoding mode that supports a bit rate of 11.0, 8.0, 6.65 or 5.8 kbps. The speech encoder 117 adaptively selects an either 8.0, 6.65, 5.8 or 4.5 kbps encoding bit rate mode when only the half rate channel has been allocated. Of course these encoding bit rates and the aforementioned channel allocations are only representative of the present embodiment. Other variations to meet the goals of alternate embodiments are contemplated.

With either the full or half rate allocation, the speech encoder 117 attempts to communicate using the highest encoding bit rate mode that the allocated channel will support. If the allocated channel is or becomes noisy or otherwise restrictive to the highest or higher encoding bit rates, the speech encoder 117 adapts by selecting a lower bit rate encoding mode.

-10-

Similarly, when the communication channel 103 becomes more favorable, the speech encoder 117 adapts by switching to a higher bit rate encoding mode.

With lower bit rate encoding, the speech encoder 117 incorporates various techniques to generate better low bit rate speech reproduction. Many of the techniques applied are based on characteristics of the speech itself. For example, with lower bit rate encoding, the speech encoder 117 classifies noise, unvoiced speech, and voiced speech so that an appropriate modeling scheme corresponding to a particular classification can be selected and implemented. Thus, the speech encoder 117 adaptively selects from among a plurality of modeling schemes those most suited for the current speech. The speech encoder 117 also applies various other techniques to optimize the modeling as set forth in more detail below.

Fig. 1b is a schematic block diagram illustrating several variations of an exemplary communication device employing the functionality of Fig. 1a. A communication device 151 comprises both a speech encoder and decoder for simultaneous capture and reproduction of speech. Typically within a single housing, the communication device 151 might, for example, comprise a cellular telephone, portable telephone, computing system, etc. Alternatively, with some modification to include for example a memory element to store encoded speech information the communication device 151 might comprise an answering machine, a recorder, voice mail system, etc.

A microphone 155 and an A/D converter 157 coordinate to deliver a digital voice signal to an encoding system 159. The encoding system 159 performs speech and channel encoding and delivers resultant speech information to the channel. The delivered speech information may be destined for another communication device (not shown) at a remote location.

As speech information is received, a decoding system 165 performs channel and speech decoding then coordinates with a D/A converter 167 and a speaker 169 to reproduce something that sounds like the originally captured speech.

The encoding system 159 comprises both a speech processing circuit 185 that performs speech encoding, and a channel processing circuit 187 that performs channel encoding. Similarly, the decoding system 165 comprises a speech processing circuit 189 that performs speech decoding, and a channel processing circuit 191 that performs channel decoding.

Although the speech processing circuit 185 and the channel processing circuit 187 are separately illustrated, they might be combined in part or in total into a single unit. For example, the speech processing circuit 185 and the channel processing circuit 187 might share a single DSP (digital signal processor) and/or other processing circuitry. Similarly, the speech processing circuit 189 and the channel processing circuit 191 might be entirely separate or combined in part or in whole. Moreover, combinations in whole or in part might be applied to the speech processing circuits 185 and 189, the channel processing circuits 187 and 191, the processing circuits 185, 187, 189 and 191, or otherwise.

The encoding system 159 and the decoding system 165 both utilize a memory 161. The speech processing circuit 185 utilizes a fixed codebook 181 and an adaptive codebook 183 of a speech memory 177 in the source encoding process. The channel processing circuit 187 utilizes a channel memory 175 to perform channel encoding. Similarly, the speech processing circuit 189 utilizes the fixed codebook 181 and the adaptive codebook 183 in the source decoding process. The channel processing circuit 187 utilizes the channel memory 175 to perform channel decoding.

Although the speech memory 177 is shared as illustrated, separate copies thereof can be assigned for the processing circuits 185 and 189. Likewise, separate channel memory can be allocated to both the processing circuits 187 and 191. The memory 161 also contains software utilized by the processing circuits 185, 187, 189 and 191 to perform various functionality required in the source and channel encoding and decoding processes.

Figs. 2-4 are functional block diagrams illustrating a multi-step encoding approach used by one embodiment of the speech encoder illustrated in Figs. 1a and 1b. In particular, Fig. 2 is a functional block diagram illustrating of a first stage of operations performed by one embodiment of the speech encoder shown in Figs. 1a and 1b. The speech encoder, which comprises encoder processing circuitry, typically operates pursuant to software instruction carrying out the following functionality.

At a block 215, source encoder processing circuitry performs high pass filtering of a speech signal 211. The filter uses a cutoff frequency of around 80 Hz to remove, for example, 60 Hz power line noise and other lower frequency signals. After such filtering, the source encoder processing circuitry applies a perceptual weighting filter as represented by a block 219. The perceptual weighting filter operates to emphasize the valley areas of the filtered speech signal.

If the encoder processing circuitry selects operation in a pitch preprocessing (PP) mode as indicated at a control block 245, a pitch preprocessing operation is performed on the weighted speech signal at a block 225. The pitch preprocessing operation involves warping the weighted speech signal to match interpolated pitch values that will be generated by the decoder processing circuitry. When pitch preprocessing is applied, the warped speech signal is designated a first target signal 229. If pitch preprocessing is not selected the control block 245, the weighted

speech signal passes through the block 225 without pitch preprocessing and is designated the first target signal 229.

As represented by a block 255, the encoder processing circuitry applies a process wherein a contribution from an adaptive codebook 257 is selected along with a corresponding gain 257 which minimize a first error signal 253. The first error signal 253 comprises the difference between the first target signal 229 and a weighted, synthesized contribution from the adaptive codebook 257.

At blocks 247, 249 and 251, the resultant excitation vector is applied after adaptive gain reduction to both a synthesis and a weighting filter to generate a modeled signal that best matches the first target signal 229. The encoder processing circuitry uses LPC (linear predictive coding) analysis, as indicated by a block 239, to generate filter parameters for the synthesis and weighting filters. The weighting filters 219 and 251 are equivalent in functionality.

Next, the encoder processing circuitry designates the first error signal 253 as a second target signal for matching using contributions from a fixed codebook 261. The encoder processing circuitry searches through at least one of the plurality of subcodebooks within the fixed codebook 261 in an attempt to select a most appropriate contribution while generally attempting to match the second target signal.

More specifically, the encoder processing circuitry selects an excitation vector, its corresponding subcodebook and gain based on a variety of factors. For example, the encoding bit rate, the degree of minimization, and characteristics of the speech itself as represented by a block 279 are considered by the encoder processing circuitry at control block 275. Although many other factors may be considered, exemplary characteristics include speech classification, noise level, sharpness, periodicity, etc. Thus, by considering other such factors, a first

subcodebook with its best excitation vector may be selected rather than a second subcodebook's best excitation vector even though the second subcodebook's better minimizes the second target signal 265.

Fig. 3 is a functional block diagram depicting of a second stage of operations performed by the embodiment of the speech encoder illustrated in Fig. 2. In the second stage, the speech encoding circuitry simultaneously uses both the adaptive and the fixed codebook vectors found in the first stage of operations to minimize a third error signal 311.

The speech encoding circuitry searches for optimum gain values for the previously identified excitation vectors (in the first stage) from both the adaptive and fixed codebooks 257 and 261. As indicated by blocks 307 and 309, the speech encoding circuitry identifies the optimum gain by generating a synthesized and weighted signal, i.e., via a block 301 and 303, that best matches the first target signal 229 (which minimizes the third error signal 311). Of course if processing capabilities permit, the first and second stages could be combined wherein joint optimization of both gain and adaptive and fixed codebook vector selection could be used.

Fig. 4 is a functional block diagram depicting of a third stage of operations performed by the embodiment of the speech encoder illustrated in Figs. 2 and 3. The encoder processing circuitry applies gain normalization, smoothing and quantization, as represented by blocks 401, 403 and 405, respectively, to the jointly optimized gains identified in the second stage of encoder processing. Again, the adaptive and fixed codebook vectors used are those identified in the first stage processing.

With normalization, smoothing and quantization functionally applied, the encoder processing circuitry has completed the modeling process. Therefore, the modeling parameters identified are communicated to the decoder. In particular, the encoder processing circuitry

-15-

delivers an index to the selected adaptive codebook vector to the channel encoder via a multiplexor 419. Similarly, the encoder processing circuitry delivers the index to the selected fixed codebook vector, resultant gains, synthesis filter parameters, etc., to the multiplexor 419. The multiplexor 419 generates a bit stream 421 of such information for delivery to the channel encoder for communication to the channel and speech decoder of receiving device.

Fig. 5 is a block diagram of an embodiment illustrating functionality of speech decoder having corresponding functionality to that illustrated in Figs. 2-4. As with the speech encoder, the speech decoder, which comprises decoder processing circuitry, typically operates pursuant to software instruction carrying out the following functionality.

A demultiplexor 511 receives a bit stream 513 of speech modeling indices from an often remote encoder via a channel decoder. As previously discussed, the encoder selected each index value during the multi-stage encoding process described above in reference to Figs. 2-4. The decoder processing circuitry utilizes indices, for example, to select excitation vectors from an adaptive codebook 515 and a fixed codebook 519, set the adaptive and fixed codebook gains at a block 521, and set the parameters for a synthesis filter 531.

With such parameters and vectors selected or set, the decoder processing circuitry generates a reproduced speech signal 539. In particular, the codebooks 515 and 519 generate excitation vectors identified by the indices from the demultiplexor 511. The decoder processing circuitry applies the indexed gains at the block 521 to the vectors which are summed. At a block 527, the decoder processing circuitry modifies the gains to emphasize the contribution of vector from the adaptive codebook 515. At a block 529, adaptive tilt compensation is applied to the combined vectors with a goal of flattening the excitation spectrum. The decoder processing circuitry performs synthesis filtering at the block 531 using the flattened excitation signal.

-16-



Finally, to generate the reproduced speech signal 539, post filtering is applied at a block 535 deemphasizing the valley areas of the reproduced speech signal 539 to reduce the effect of distortion.

In the exemplary cellular telephony embodiment of the present invention, the A/D converter 115 (Fig. 1a) will generally involve analog to uniform digital PCM including: 1) an input level adjustment device; 2) an input anti-aliasing filter; 3) a sample-and-hold device sampling at 8 kHz; and 4) analog to uniform digital conversion to 13-bit representation.

Similarly, the D/A converter 135 will generally involve uniform digital PCM to analog including: 1) conversion from 13-bit/8 kHz uniform PCM to analog; 2) a hold device; 3)

reconstruction filter including  $x/\sin(x)$  correction; and 4) an output level adjustment device.

In terminal equipment, the A/D function may be achieved by direct conversion to 13-bit uniform PCM format, or by conversion to 8-bit/A-law compounded format. For the D/A operation, the inverse operations take place.

The encoder 117 receives data samples with a resolution of 13 bits left justified in a 16-bit word. The three least significant bits are set to zero. The decoder 133 outputs data in the same format. Outside the speech codec, further processing can be applied to accommodate traffic data having a different representation.

A specific embodiment of an AMR (adaptive multi-rate) codec with the operational functionality illustrated in Figs. 2-5 uses five source codecs with bit-rates 11.0, 8.0, 6.65, 5.8 and 4.55 kbps. Four of the highest source coding bit-rates are used in the full rate channel and the four lowest bit-rates in the half rate channel.

All five source codecs within the AMR codec are generally based on a code-excited linear predictive (CELP) coding model. A 10th order linear prediction (LP), or short-term,

synthesis filter, e.g., used at the blocks 249, 267, 301, 407 and 531 (of Figs. 2-5), is used which is given by:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{i=1}^m \hat{a}_i z^{-i}} \quad (1)$$

where  $\hat{a}_i, i = 1, \dots, m$ , are the (quantized) linear prediction (LP) parameters.

A long-term filter, i.e., the pitch synthesis filter, is implemented using the either an adaptive codebook approach or a pitch pre-processing approach. The pitch synthesis filter is given by:

$$\frac{1}{B(z)} = \frac{1}{1 - g_p z^{-T}} \quad (2)$$

where  $T$  is the pitch delay and  $g_p$  is the pitch gain.

With reference to Fig. 2, the excitation signal at the input of the short-term LP synthesis filter at the block 249 is constructed by adding two excitation vectors from the adaptive and the fixed codebooks 257 and 261, respectively. The speech is synthesized by feeding the two properly chosen vectors from these codebooks through the short-term synthesis filter at the block 249 and 267, respectively.

The optimum excitation sequence in a codebook is chosen using an analysis-by-synthesis search procedure in which the error between the original and synthesized speech is minimized according to a perceptually weighted distortion measure. The perceptual weighting filter, e.g., at the blocks 251 and 268, used in the analysis-by-synthesis search technique is given by:

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)} \quad (3)$$

where  $A(z)$  is the unquantized LP filter and  $0 < \gamma_2 < \gamma_1 \leq 1$  are the perceptual weighting factors. The values  $\gamma_1 = [0.9, 0.94]$  and  $\gamma_2 = 0.6$  are used. The weighting filter, e.g., at the

blocks 251 and 268, uses the unquantized LP parameters while the formant synthesis filter, e.g., at the blocks 249 and 267, uses the quantized LP parameters. Both the unquantized and quantized LP parameters are generated at the block 239.

The present encoder embodiment operates on 20 ms (millisecond) speech frames corresponding to 160 samples at the sampling frequency of 8000 samples per second. At each 160 speech samples, the speech signal is analyzed to extract the parameters of the CELP model, i.e., the LP filter coefficients, adaptive and fixed codebook indices and gains. These parameters are encoded and transmitted. At the decoder, these parameters are decoded and speech is synthesized by filtering the reconstructed excitation signal through the LP synthesis filter.

More specifically, LP analysis at the block 239 is performed twice per frame but only a single set of LP parameters is converted to line spectrum frequencies (LSF) and vector quantized using predictive multi-stage quantization (PMVQ). The speech frame is divided into subframes. Parameters from the adaptive and fixed codebooks 257 and 261 are transmitted every subframe. The quantized and unquantized LP parameters or their interpolated versions are used depending on the subframe. An open-loop pitch lag is estimated at the block 241 once or twice per frame for PP mode or LTP mode, respectively.

Each subframe, at least the following operations are repeated. First, the encoder processing circuitry (operating pursuant to software instruction) computes  $x(n)$ , the first target signal 229, by filtering the LP residual through the weighted synthesis filter  $W(z)H(z)$  with the initial states of the filters having been updated by filtering the error between LP residual and excitation. This is equivalent to an alternate approach of subtracting the zero input response of the weighted synthesis filter from the weighted speech signal.

Second, the encoder processing circuitry computes the impulse response,  $h(n)$ , of the weighted synthesis filter. Third, in the LTP mode, closed-loop pitch analysis is performed to find the pitch lag and gain, using the first target signal 229,  $x(n)$ , and impulse response,  $h(n)$ , by searching around the open-loop pitch lag. Fractional pitch with various sample resolutions are used.

In the PP mode, the input original signal has been pitch-preprocessed to match the interpolated pitch contour, so no closed-loop search is needed. The LTP excitation vector is computed using the interpolated pitch contour and the past synthesized excitation.

Fourth, the encoder processing circuitry generates a new target signal  $x_1(n)$ , the second target signal 253, by removing the adaptive codebook contribution (filtered adaptive code vector) from  $x(n)$ . The encoder processing circuitry uses the second target signal 253 in the fixed codebook search to find the optimum innovation.

Fifth, for the 11.0 kbps bit rate mode, the gains of the adaptive and fixed codebook are scalar quantized with 4 and 5 bits respectively (with moving average prediction applied to the fixed codebook gain). For the other modes the gains of the adaptive and fixed codebook are vector quantized (with moving average prediction applied to the fixed codebook gain).

Finally, the filter memories are updated using the determined excitation signal for finding the first target signal in the next subframe.

The bit allocation of the AMR codec modes is shown in table 1. For example, for each 20 ms speech frame, 220, 160, 133, 116 or 91 bits are produced, corresponding to bit rates of 11.0, 8.0, 6.65, 5.8 or 4.55 kbps, respectively.

Table 1: Bit allocation of the AMR coding algorithm for 20 ms frame

CODING RATE	11.0KBPS	8.0KBPS	6.65KBPS	5.40KBPS	4.55KBPS
Frame size	20ms	20ms	20ms	20ms	20ms
LTP order	10 <sup>th</sup> order	10 <sup>th</sup> order	10 <sup>th</sup> order	10 <sup>th</sup> order	10 <sup>th</sup> order
Predictor for LSF	1 predictor, 0 bit/frame	1 predictor, 0 bit/frame	1 predictor, 0 bit/frame	1 predictor, 0 bit/frame	1 predictor, 0 bit/frame
Quantization	24 bit/frame	24 bit/frame	24 bit/frame	24 bit/frame	24 bit/frame
LSF Quantization	2 bit/frame	2 bit/frame	2 bit/frame	2 bit/frame	2 bit/frame
LTP Interpolation	0 bit	0 bit	0 bit	0 bit	0 bit
Coding mode bit	LTP	LTP	LTP	LTP	LTP
Pitch mode	LTP	LTP	LTP	LTP	LTP
Subframe size	30 bit/frame (60ms)	30 bit/frame (60ms)	30 bit/frame (60ms)	30 bit/frame (60ms)	30 bit/frame (60ms)
Pitch LTP	31 bit/frame	31 bit/frame	31 bit/frame	31 bit/frame	31 bit/frame
Pitch excitation	9 bit/frame	9 bit/frame	9 bit/frame	9 bit/frame	9 bit/frame
Gain quantization	9 bit/frame	9 bit/frame	9 bit/frame	9 bit/frame	9 bit/frame
Total	160	160	160	160	160

With reference to Fig. 5, the decoder processing circuitry, pursuant to software control,

reconstructs the speech signal using the transmitted modeling indices extracted from the received bit stream by the demultiplexor 511. The decoder processing circuitry decodes the indices to obtain the coder parameters at each transmission frame. These parameters are the LSF vectors, the fractional pitch lags, the innovative code vectors, and the two gains.

The LSF vectors are converted to the LP filter coefficients and interpolated to obtain LP filters at each subframe. At each subframe, the decoder processing circuitry constructs the excitation signal by: 1) identifying the adaptive and innovative code vectors from the codebooks 515 and 519; 2) scaling the contributions by their respective gains at the block 521; 3) summing the scaled contributions; and 3) modifying and applying adaptive tilt compensation at the blocks 527 and 529. The speech signal is also reconstructed on a subframe basis by filtering the excitation through the LP synthesis at the block 531. Finally, the speech signal is passed through an adaptive post filter at the block 535 to generate the reproduced speech signal 539.

The AMR encoder will produce the speech modeling information in a unique sequence and format, and the AMR decoder receives the same information in the same way. The different parameters of the encoded speech and their individual bits have unequal importance with respect

to subjective quality. Before being submitted to the channel encoding function the bits are rearranged in the sequence of importance.

Two pre-processing functions are applied prior to the encoding process: high-pass filtering and signal down-scaling. Down-scaling consists of dividing the input by a factor of 2 to reduce the possibility of overflows in the fixed point implementation. The high-pass filtering at the block 215 (Fig. 2) serves as a precaution against undesired low frequency components. A filter with cut off frequency of 80 Hz is used, and it is given by:

$$H_u(z) = \frac{0.92727435 - 1.8544941z^{-1} + 0.92727435z^{-2}}{1 - 1.9059465z^{-1} + 0.9114024z^{-2}}$$

Down scaling and high-pass filtering are combined by dividing the coefficients of the numerator of  $H_u(z)$  by 2.

Short-term prediction, or linear prediction (LP) analysis is performed twice per speech frame using the autocorrelation approach with 30 ms windows. Specifically, two LP analyses are performed twice per frame using two different windows. In the first LP analysis (LP\_analysis\_1), a hybrid window is used which has its weight concentrated at the fourth subframe. The hybrid window consists of two parts. The first part is half a Hamming window, and the second part is a quarter of a cosine cycle. The window is given by:

$$w_1(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{m}{L}\right) & n = 0 \text{ to } 214, L = 215 \\ \cos\left(\frac{0.49(n-L)\pi}{25}\right) & n = 215 \text{ to } 239 \end{cases}$$

In the second LP analysis (LP\_analysis\_2), a symmetric Hamming window is used.

$$w_2(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{\pi n}{L}\right) & n = 0 \text{ to } 119, L = 120 \\ 0.54 + 0.46 \cos\left(\frac{(n-L)\pi}{120}\right) & n = 120 \text{ to } 239 \end{cases}$$

past frame	current frame	future frame
55	160	25 (samples)

In either LP analysis, the autocorrelations of the windowed speech  $s'(n)$ ,  $n = 0, 239$  are computed by:

$$r(k) = \sum_{n=k}^{239} s'(n) s'(n-k), \quad k = 0, 10.$$

A 60 Hz bandwidth expansion is used by lag windowing, the autocorrelations using the window:

$$w_{eq}(i) = \exp\left[-\frac{1}{2} \left(\frac{2\pi 60i}{8000}\right)^2\right], \quad i = 1, 10.$$

Moreover,  $r(0)$  is multiplied by a white noise correction factor 1.0001 which is equivalent to adding a noise floor at -40 dB.

The modified autocorrelations  $r'(0) = 1.0001 r(0)$  and  $r'(k) = r(k) w_{eq}(k)$ ,  $k = 1, 10$  are

used to obtain the reflection coefficients  $k_i$  and LP filter coefficients  $a_i$ ,  $i = 1, 10$  using the Levinson-Durbin algorithm. Furthermore, the LP filter coefficients  $a_i$  are used to obtain the Line Spectral Frequencies (LSFs).

The interpolated unquantized LP parameters are obtained by interpolating the LSF coefficients obtained from the LP analysis\_1 and those from LP\_analysis\_2 as:

$$\begin{aligned} q_1(n) &= 0.5q_1(n-1) + 0.5q_2(n) \\ q_3(n) &= 0.5q_2(n) + 0.5q_4(n) \end{aligned}$$

where  $q_1(n)$  is the interpolated LSF for subframe 1,  $q_3(n)$  is the LSF of subframe 2 obtained from LP\_analysis\_2 of current frame,  $q_1(n)$  is the interpolated LSF for subframe 3,  $q_4(n-1)$  is the LSF (cosine domain) from LP\_analysis\_1 of previous frame, and  $q_4(n)$  is the LSF for subframe 4 obtained from LP\_analysis\_1 of current frame. The interpolation is carried out in the cosine domain.

A VAD (Voice Activity Detection) algorithm is used to classify input speech frames into either active voice or inactive voice frame (background noise or silence) at a block 235 (Fig. 2).

The input speech  $s(n)$  is used to obtain a weighted speech signal  $s_w(n)$  by passing  $s(n)$  through a filter:

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)}.$$

That is, in a subframe of size  $L_{SF}$ , the weighted speech is given by:

$$s_w(n) = s(n) + \sum_{i=1}^{10} a_i \gamma_1^i s(n-i) - \sum_{i=1}^{10} a_i \gamma_2^i s_w(n-i), \quad n = 0, L_{SF}-1.$$

A voiced/unvoiced classification and mode decision within the block 279 using the input

speech  $s(n)$  and the residual  $r_w(n)$  is derived where:

$$r_w(n) = s(n) + \sum_{i=1}^{10} a_i \gamma_1^i s(n-i), \quad n = 0, L_{SF}-1.$$

The classification is based on four measures: 1) speech sharpness  $P1\_SHP$ ; 2) normalized one delay correlation  $P2\_R1$ ; 3) normalized zero-crossing rate  $P3\_ZC$ ; and 4) normalized LP residual energy  $P4\_RE$ .

The speech sharpness is given by:

$$P1\_SHP = \frac{\sum_{n=0}^L abs(r_n(n))}{MaxL}$$

where  $Max$  is the maximum of  $abs(r_n(n))$  over the specified interval of length  $L$ . The

normalized one delay correlation and normalized zero-crossing rate are given by:

$$P2\_RI = \frac{\sum_{n=0}^{L-1} s(n)s(n+1)}{\sqrt{\sum_{n=0}^{L-1} s(n)s(n) \sum_{n=0}^{L-1} s(n+1)s(n+1)}}$$

$$P3\_ZC = \frac{1}{2L} \sum_{i=1}^{L-1} |sgn[s(i)] - sgn[s(i-1)]|,$$

where  $sgn$  is the sign function whose output is either 1 or -1 depending that the input sample is positive or negative. Finally, the normalized LP residual energy is given by:

$$P4\_RE = 1 - \sqrt{lpc\_gain}$$

where  $lpc\_gain = \prod_{i=1}^{10} (1 - k_i^2)$ , where  $k_i$  are the reflection coefficients obtained from LP analysis\_1.

The voiced/unvoiced decision is derived if the following conditions are met:

if  $P2\_RI < 0.6$  and  $P1\_SHP > 0.2$  set mode = 2,  
 if  $P3\_ZC > 0.4$  and  $P1\_SHP > 0.18$  set mode = 2,  
 if  $P4\_RE < 0.4$  and  $P1\_SHP > 0.2$  set mode = 2,  
 if  $(P2\_RI < -1.2 + 3.2P1\_SHP)$  set VUV = -3  
 if  $(P4\_RE < -0.21 + 1.4286P1\_SHP)$  set VUV = -3  
 if  $(P3\_ZC > 0.8 - 0.6P1\_SHP)$  set VUV = -3  
 if  $(P4\_RE < 0.1)$  set VUV = -3

Open loop pitch analysis is performed once or twice (each 10 ms) per frame depending on the coding rate in order to find estimates of the pitch lag at the block 241 (Fig. 2). It is based

-25-

on the weighted speech signal  $s_w(n+n_m)$ ,  $n=0,1,\dots,79$ , in which  $n_m$  defines the location of this signal on the first half frame or the last half frame. In the first step, four maxima of the

correlation:

$$C_k = \sum_{n=0}^{79} s_w(n_m+n)s_w(n_m+n-k)$$

are found in the four ranges 17....33, 34....67, 68....135, 136....145, respectively. The retained maxima  $C_{k_i}$ ,  $i=1,2,3,4$ , are normalized by dividing by:

$$\sqrt{\sum_{n=0}^{79} s_w^2(n_m+n-k)}, \quad i=1,\dots,4, \text{ respectively.}$$

The normalized maxima and corresponding delays are denoted by  $(R_i, k_i)$ ,  $i=1,2,3,4$ .

In the second step, a delay,  $k_i$  among the four candidates, is selected by maximizing the four normalized correlations. In the third step,  $k_i$  is probably corrected to  $k_i(i \leq D)$  by favoring the lower ranges. That is,  $k_i(i \leq D)$  is selected if  $k_i$  is within  $(k/m-4, k/m+4)$ ,  $m=2,3,4,5$ , and if  $k_i > k_j$ ,  $0.95^{i-j} D$ ,  $i < j$ , where  $D$  is 1.0, 0.85, or 0.65, depending on whether the previous frame is unvoiced, the previous frame is voiced and  $k_i$  is in the neighborhood (specified by  $\pm 8$ ) of the previous pitch lag, or the previous two frames are voiced and  $k_i$  is in the neighborhood of the previous two pitch lags. The final selected pitch lag is denoted by  $T_{op}$ .

A decision is made every frame to either operate the LTP (long-term prediction) as the traditional CELP approach (LTP\_mode=1), or as a modified time warping approach (LTP\_mode=0) herein referred to as PP (pitch preprocessing). For 4.55 and 5.8 kbps encoding bit rates, LTP\_mode is set to 0 at all times. For 8.0 and 11.0 kbps, LTP\_mode is set to 1 all of the time. Whereas, for a 6.65 kbps encoding bit rate, the encoder decides whether to operate in the LTP or PP mode. During the PP mode, only one pitch lag is transmitted per coding frame.

-26-

For 6.65 kbps, the decision algorithm is as follows. First, at the block 241, a prediction of the pitch lag  $pit$  for the current frame is determined as follows:

```

if (LTP_MODE_m == 1)
    pit = lagl1 + 2.4*(lag_f3)-lagl1);
else
    pit = lag_f1) + 2.75*(lag_f3)-lag_f1);

```

where  $LTP\_mod\ e\_m$  is previous frame  $LTP\_mod\ e$ ,  $lag\_f1$ ,  $lag\_f3$  are the past closed loop pitch lags for second and fourth subframes respectively,  $lagl1$  is the current frame open-loop pitch lag at the second half of the frame, and,  $lagl1$  is the previous frame open-loop pitch lag at the first half of the frame.

Second, a normalized spectrum difference between the Line Spectrum Frequencies (LSF) of current and previous frame is computed as:

```

e_lsf = 1/10 * sum(abs(LSF(i) - LSF_m(i)), i)
if (abs(pit-lagf) < TH and abs(lag_f3-lagf) < lagf*0.2)
    if (Rp > 0.5 && pgain_past > 0.7 and e_lsf < 0.5/30) LTP_mod_e = 0;
else LTP_mod_e = 1;

```

where  $Rp$  is current frame normalized pitch correlation,  $pgain\_past$  is the quantized pitch gain from the fourth subframe of the past frame,  $TH = MIN(lagf*0.1, 5)$ , and  $TH = MAX(2.0, TH)$ .

The estimation of the precise pitch lag at the end of the frame is based on the normalized correlation:

$$R_y = \frac{\sum_{n=0}^L s_u(n+nl) s_u(n+nl-k)}{\sqrt{\sum_{n=0}^L s_u^2(n+nl-k)}}$$

where  $s_u(n+nl)$ ,  $n = 0, 1, \dots, L-1$ , represents the last segment of the weighted speech signal including the look-ahead (the look-ahead length is 25 samples), and the size  $L$  is defined according to the open-loop pitch lag  $T_{op}$  with the corresponding normalized correlation  $C_{r_u}$ :

```

if (C_r_u > 0.6)
    L = max(50, T_op)
    L = min(80, L)
else
    L = 80

```

In the first step, one integer lag  $k$  is selected maximizing the  $R_k$  in the range  $k \in [T_{op} - 10, T_{op} + 10]$  bounded by [17, 145]. Then, the precise pitch lag  $P_m$  and the

corresponding index  $I_m$  for the current frame is searched around the integer lag,  $[k-1, k+1]$ , by up-sampling  $R_k$ .

The possible candidates of the precise pitch lag are obtained from the table named as *PitLagTab8b(i)*,  $i=0, 1, \dots, 127$ . In the last step, the precise pitch lag  $P_m = PitLagTab8b(I_m)$  is possibly modified by checking the accumulated delay  $\tau_{acc}$  due to the modification of the speech signal:

```

if (tau_acc > 5) I_m = min(I_m + 1, 127); and
if (tau_acc < -5) I_m = max(I_m - 1, 0);

```

The precise pitch lag could be modified again:

```

if (tau_acc > 10) I_m = min(I_m + 1, 127); and
if (tau_acc < -10) I_m = max(I_m - 1, 0);

```

The obtained index  $I_m$  will be sent to the decoder.

The pitch lag contour,  $\tau_c(n)$ , is defined using both the current lag  $P_m$  and the previous

lag  $P_{m-1}$ :

if (  $|P_n - P_{m-1}| < 0.2 \min(P_n, P_{m-1})$  )  
 $\tau_c(n) = P_{m-1} + n(P_m - P_{m-1}) / L_f, \quad n = 0, 1, \dots, L_f - 1$   
 else  
 $\tau_c(n) = P_m, \quad n = L_f, \dots, 170$   
 $\tau_c(n) = P_{m-1}, \quad n = 0, 1, \dots, 39;$   
 $\tau_c(n) = P_m, \quad n = 40, \dots, 170$

where  $L_f = 160$  is the frame size.

One frame is divided into 3 subframes for the long-term preprocessing. For the first two subframes, the subframe size,  $L_n$ , is 53, and the subframe size for searching,  $L_n$ , is 70. For the last subframe,  $L_n$  is 54 and  $L_n$  is:

$$L_n = \min(70, L_f + L_{n-1} - 10 - \tau_{acc}).$$

where  $L_{n-1} = 25$  is the look-ahead and the maximum of the accumulated delay  $\tau_{acc}$  is limited to 14.

The target for the modification process of the weighted speech temporally memorized in buffer,  $\hat{s}_w(m0+n)$ ,  $n = 0, 1, \dots, L_p - 1$  is calculated by warping the past modified weighted speech buffer,  $\hat{s}_w(m0+n)$ ,  $n < 0$ , with the pitch lag contour,  $\tau_c(n+m \cdot L_f)$ ,  $m = 0, 1, 2$ .

$$\hat{s}_w(m0+n) = \sum_{k=-f}^f \hat{s}_w(m0+n - T_c(n) + f) I_f(k, T_{KC}(n)), \quad n = 0, 1, \dots, L_p - 1,$$

where  $T_c(n)$  and  $T_{KC}(n)$  are calculated by:

$$T_c(n) = \text{trunc}(\tau_c(n + m \cdot L_f)),$$

$$T_{KC}(n) = \tau_c(n) - T_c(n).$$

$m$  is subframe number,  $I_f(k, T_{KC}(n))$  is a set of interpolation coefficients, and  $f$  is 10. Then, the target for matching,  $\hat{s}_f(n)$ ,  $n = 0, 1, \dots, L_p - 1$ , is calculated by weighting

$\hat{s}_w(m0+n)$ ,  $n = 0, 1, \dots, L_p - 1$ , in the time domain:

$$\hat{s}_f(n) = n \cdot \hat{s}_w(m0+n) / L_f, \quad n = 0, 1, \dots, L_f - 1,$$

$$\hat{s}_f(n) = \hat{s}_w(m0+n), \quad n = L_f, \dots, L_p - 1$$

The local integer shifting range  $[SRO, SRI]$  for searching for the best local delay is computed as the following:

if speech is unvoiced

$$SRO = -1,$$

$$SRI = L,$$

else

$$SRO = \text{round}(4 \min(1.0, \max(0.0, 1 - 0.4(P_n - 0.2))))),$$

$$SRI = \text{round}(4 \min(1.0, \max(0.0, 1 - 0.4(P_n - 0.2))))).$$

where  $P_n = \max(P_{n1}, P_{n2})$ ,  $P_{n1}$  is the average to peak ratio (i.e., sharpness) from the target signal:

$$P_{n1} = \frac{\sum_{n=0}^{L_p-1} |\hat{s}_w(m0+n)|}{L_p \max(|\hat{s}_w(m0+n)|, n = 0, 1, \dots, L_p - 1)}$$

and  $P_{n2}$  is the sharpness from the weighted speech signal:

$$P_{n2} = \frac{\sum_{n=0}^{L_p-L_f/2-1} |\hat{s}_w(n+n0+L_f/2)|}{(L_p - L_f/2) \max(|\hat{s}_w(n+n0+L_f/2)|, n = 0, 1, \dots, L_p - L_f/2 - 1)}$$

where  $n0 = \text{trunc}(m0 + \tau_{acc} + 0.5)$  (here,  $m$  is subframe number and  $\tau_{acc}$  is the previous accumulated delay).

In order to find the best local delay,  $\tau_{opt}$ , at the end of the current processing subframe, a normalized correlation vector between the original weighted speech signal and the modified matching target is defined as:

$$R_f(k) = \frac{\sum_{n=0}^{L_p-1} \hat{s}_w(n0+n+k) \hat{s}_f(n)}{\sqrt{\sum_{n=0}^{L_p-1} \hat{s}_w^2(n0+n+k) \sum_{n=0}^{L_p-1} \hat{s}_f^2(n)}}$$

A best local delay in the integer domain,  $k_{opt}$ , is selected by maximizing  $R_d(k)$  in the range of  $k \in [SRO, SRI]$ , which is corresponding to the real delay:

$$k_r = k_{opt} + n0 - m0 - \tau_{acc}$$

If  $R_d(k_{opt}) < 0.5$ ,  $k_r$  is set to zero.

In order to get a more precise local delay in the range  $[k-0.75+0.1j, j=0,1,...,15]$  around  $k_r$ ,  $R_d(k)$  is interpolated to obtain the fractional correlation vector,  $R_f(j)$ , by:

$$R_f(j) = \sum_{i=0}^{15} R_d(k_{opt} + I_j + i) I_f(i, j), \quad j = 0, 1, \dots, 15,$$

where  $\{I_f(i, j)\}$  is a set of interpolation coefficients. The optimal fractional delay index,  $j_{opt}$ , is selected by maximizing  $R_f(j)$ . Finally, the best local delay,  $\tau_{opt}$ , at the end of the current processing subframe, is given by,

$$\tau_{opt} = k_r - 0.75 + 0.1 j_{opt}$$

The local delay is then adjusted by:

$$\tau_{opt} = \begin{cases} 0, & \text{if } \tau_{acc} + \tau_{opt} > 14 \\ \tau_{opt}, & \text{otherwise} \end{cases}$$

The modified weighted speech of the current subframe, memorized in

$(\hat{s}_n(m0+n), n=0,1,\dots,L_f-1)$  to update the buffer and produce the second target signal 253 for searching the fixed codebook 261, is generated by warping the original weighted speech  $(s_n(n))$  from the original time region,

$$(m0 + \tau_{acc}, m0 + \tau_{acc} + L_f + \tau_{opt}),$$

to the modified time region,

$$(m0, m0 + L_f);$$

-31-

$$\hat{s}_n(m0+n) = \sum_{i=-J_f+1}^{J_f} \hat{s}_n(m0+n+T_w(n)+i) I_f(i, T_w(n)), \quad n=0,1,\dots,L_f-1,$$

where  $T_w(n)$  and  $T_w(n)$  are calculated by:

$$\begin{aligned} T_w(n) &= \text{trunc}(\tau_{acc} + n \cdot \tau_{opt} / L_f), \\ T_w(n) &= \tau_{acc} + n \cdot \tau_{opt} / L_f - T_w(n), \end{aligned}$$

$\{I_f(i, T_w(n))\}$  is a set of interpolation coefficients.

After having completed the modification of the weighted speech for the current subframe,

the modified target weighted speech buffer is updated as follows:

$$\hat{s}_n(n) \leftarrow \hat{s}_n(n + L_f), \quad n=0,1,\dots,n_m-1.$$

The accumulated delay at the end of the current subframe is renewed by:

$$\tau_{acc} \leftarrow \tau_{acc} + \tau_{opt}.$$

Prior to quantization the LSFs are smoothed in order to improve the perceptual quality.

In principle, no smoothing is applied during speech and segments with rapid variations in the spectral envelope. During non-speech with slow variations in the spectral envelope, smoothing is applied to reduce unwanted spectral variations. Unwanted spectral variations could typically occur due to the estimation of the LPC parameters and LSF quantization. As an example, in stationary noise-like signals with constant spectral envelope introducing even very small variations in the spectral envelope is picked up easily by the human ear and perceived as an annoying modulation.

The smoothing of the LSFs is done as a running mean according to:

$$lsf_i(n) = \beta(n) \cdot lsf_i(n-1) + (1-\beta(n)) \cdot lsf_{est,i}(n), \quad i=1,\dots,10$$

-32-



where  $lsf\_est_i(n)$  is the  $i^{th}$  estimated LSF of frame  $n$ , and  $lsf_i(n)$  is the  $i^{th}$  LSF for quantization of frame  $n$ . The parameter  $\beta(n)$  controls the amount of smoothing, e.g. if  $\beta(n)$  is zero no smoothing is applied.

$\beta(n)$  is calculated from the VAD information (generated at the block 235) and two estimates of the evolution of the spectral envelope. The two estimates of the evolution are defined as:

$$\Delta SP = \sum_{i=1}^{10} (lsf\_est_i(n) - lsf\_est_i(n-1))^2$$

$$\Delta SP_{sm} = \sum_{i=1}^{10} (lsf\_est_i(n) - ma\_lsf_i(n-1))^2$$

$$ma\_lsf_i(n) = \beta(n) \cdot ma\_lsf_i(n-1) + (1 - \beta(n)) \cdot lsf\_est_i(n), \quad i = 1, \dots, 10$$

The parameter  $\beta(n)$  is controlled by the following logic:

Step 1:

```

if (Vad = 1 | PastVad = 1 |  $k_1 > 0.5$ )
   $N_{mode\_sm}(n-1) = 0$ 
   $\beta(n) = 0.0$ 
elseif ( $N_{mode\_sm}(n-1) > 0$  & ( $\Delta SP > 0.0015$  |  $\Delta SP_{sm} > 0.0024$ ))
   $N_{mode\_sm}(n-1) = 0$ 
   $\beta(n) = 0.0$ 
elseif ( $N_{mode\_sm}(n-1) > 1$  &  $\Delta SP > 0.0025$ )
   $N_{mode\_sm}(n-1) = 1$ 
endif

```

Step 2:

```

if (Vad = 0 & PastVad = 0)
   $N_{mode\_sm}(n) = N_{mode\_sm}(n-1) + 1$ 
  if ( $N_{mode\_sm}(n) > 5$ )
     $N_{mode\_sm}(n) = 5$ 
  endif
   $\beta(n) = \frac{0.9}{16} \cdot (N_{mode\_sm}(n) - 1)^2$ 
else
   $N_{mode\_sm}(n) = N_{mode\_sm}(n-1)$ 
endif

```

where  $k_1$  is the first reflection coefficient.

In step 1, the encoder processing circuitry checks the VAD and the evolution of the spectral envelope, and performs a full or partial reset of the smoothing if required. In step 2, the encoder processing circuitry updates the counter,  $N_{mode\_sm}(n)$ , and calculates the smoothing parameter,  $\beta(n)$ . The parameter  $\beta(n)$  varies between 0.0 and 0.9, being 0.0 for speech, music.

tonal-like signals, and non-stationary background noise and ramping up towards 0.9 when stationary background noise occurs.

The LSFs are quantized once per 20 ms frame using a predictive multi-stage vector quantization. A minimal spacing of 50 Hz is ensured between each two neighboring LSFs before quantization. A set of weights is calculated from the LSFs, given by  $w_i = K|P(f_i)|^{p_i}$  where  $f_i$  is the  $i^{\text{th}}$  LSF value and  $P(f_i)$  is the LPC power spectrum at  $f_i$  ( $K$  is an irrelevant multiplicative constant). The reciprocal of the power spectrum is obtained by (up to a multiplicative constant):

$$P(f_i)^{-1} = \begin{cases} (1 - \cos(2\pi f_i)) \prod_{j=1}^M [\cos(2\pi f_j) - \cos(2\pi f_i)]^2 & \text{even } i \\ (1 + \cos(2\pi f_i)) \prod_{j=1}^M [\cos(2\pi f_j) - \cos(2\pi f_i)]^2 & \text{odd } i \end{cases}$$

and the power of  $-0.4$  is then calculated using a lookup table and cubic-spline interpolation between table entries.

A vector of mean values is subtracted from the LSFs, and a vector of prediction error vector  $f_e$  is calculated from the mean removed LSFs vector, using a full-matrix AR(2) predictor. A single predictor is used for the rates 5.8, 6.65, 8.0, and 11.0 kbps coders, and two sets of prediction coefficients are tested as possible predictors for the 4.55 kbps coder.

The vector of prediction error is quantized using a multi-stage VQ, with multi-surviving candidates from each stage to the next stage. The two possible sets of prediction error vectors generated for the 4.55 kbps coder are considered as surviving candidates for the first stage.

The first 4 stages have 64 entries each, and the fifth and last table have 16 entries. The first 3 stages are used for the 4.55 kbps coder, the first 4 stages are used for the 5.8, 6.65 and 8.0 kbps coders, and all 5 stages are used for the 11.0 kbps coder. The following table summarizes the number of bits used for the quantization of the LSFs for each rate.

	prediction	1 <sup>st</sup> stage	2 <sup>nd</sup> stage	3 <sup>rd</sup> stage	4 <sup>th</sup> stage	5 <sup>th</sup> stage	total
4.55 kbps	1	6	6	6			19
5.8 kbps	0	6	6	6	6		24
6.65 kbps	0	6	6	6	6		24
8.0 kbps	0	6	6	6	6		24
11.0 kbps	0	6	6	6	6	4	28

The number of surviving candidates for each stage is summarized in the following table.

	prediction candidates into the 1 <sup>st</sup> stage	Surviving candidates from the 1 <sup>st</sup> stage	surviving candidates from the 2 <sup>nd</sup> stage	surviving candidates from the 3 <sup>rd</sup> stage	surviving candidates from the 4 <sup>th</sup> stage
4.55 kbps	2	10	6	4	
5.8 kbps	1	8	6	4	
6.65 kbps	1	8	8	4	
8.0 kbps	1	8	8	4	
11.0 kbps	1	8	6	4	4

The quantization in each stage is done by minimizing the weighted distortion measure given by:

$$e_k = \sum_{i=1}^I w_i (f_i - c_i^k)^2$$

The code vector with index  $k_{\min}$  which minimizes  $e_k$  such that  $e_{k_{\min}} < e_k$  for all  $k$ , is chosen to represent the prediction/quantization error ( $f_e$  represents in this equation both the initial prediction error to the first stage and the successive quantization error from each stage to the next one).

The final choice of vectors from all of the surviving candidates (and for the 4.55 kbps coder - also the predictor) is done at the end, after the last stage is searched, by choosing a

combined set of vectors (and predictor) which minimizes the total error. The contribution from all of the stages is summed to form the quantized prediction error vector, and the quantized prediction error is added to the prediction states and the mean LSFs value to generate the quantized LSFs vector.

For the 4.55 kbps coder, the number of order flips of the LSFs as the result of the quantization is counted, and if the number of flips is more than 1, the LSFs vector is replaced with 0.9 · (LSFs of previous frame) + 0.1 · (mean LSFs value). For all the rates, the quantized LSFs are ordered and spaced with a minimal spacing of 50 Hz.

The interpolation of the quantized LSF is performed in the cosine domain in two ways depending on the LTP\_mode. If the LTP\_mode is 0, a linear interpolation between the quantized LSF set of the current frame and the quantized LSF set of the previous frame is performed to get the LSF set for the first, second and third subframes as:

$$\begin{aligned}\bar{q}_1(n) &= 0.75\bar{q}_4(n-1) + 0.25\bar{q}_4(n) \\ \bar{q}_2(n) &= 0.5\bar{q}_4(n-1) + 0.5\bar{q}_4(n) \\ \bar{q}_3(n) &= 0.25\bar{q}_4(n-1) + 0.75\bar{q}_4(n)\end{aligned}$$

where  $\bar{q}_4(n-1)$  and  $\bar{q}_4(n)$  are the cosines of the quantized LSF sets of the previous and current frames, respectively, and  $\bar{q}_1(n)$ ,  $\bar{q}_2(n)$  and  $\bar{q}_3(n)$  are the interpolated LSF sets in cosine domain for the first, second and third subframes respectively.

If the LTP\_mode is 1, a search of the best interpolation path is performed in order to get the interpolated LSF sets. The search is based on a weighted mean absolute difference between a reference LSF set  $\bar{r}(n)$  and the LSF set obtained from LP analysis\_2  $\bar{l}(n)$ . The weights  $\bar{w}$  are computed as follows:

$$\begin{aligned}w(0) &= (1 - l(0))(1 - l(1) + l(0)) \\ w(9) &= (1 - l(9))(1 - l(9) + l(8)) \\ \text{for } i &= 1 \text{ to } 9\end{aligned}$$

$$w(i) = (1 - l(i))(1 - Min(l(i+1) - l(i), l(i) - l(i-1)))$$

where  $Min(a, b)$  returns the smallest of a and b.

There are four different interpolation paths. For each path, a reference LSF set  $\bar{r}(n)$  in cosine domain is obtained as follows:

$$\bar{r}(n) = \alpha(k)\bar{q}_4(n) + (1 - \alpha(k))\bar{q}_4(n-1), k = 1 \text{ to } 4$$

$\bar{\alpha} = \{0.4, 0.5, 0.6, 0.7\}$  for each path respectively. Then the following distance measure is

computed for each path as:

$$D = |\bar{r}(n) - \bar{l}(n)|^T \bar{w}$$

The path leading to the minimum distance D is chosen and the corresponding reference LSF set

$\bar{r}(n)$  is obtained as:

$$\bar{r}(n) = \alpha_{\text{opt}}\bar{q}_4(n) + (1 - \alpha_{\text{opt}})\bar{q}_4(n-1)$$

The interpolated LSF sets in the cosine domain are then given by:

$$\begin{aligned}\bar{q}_1(n) &= 0.5\bar{q}_4(n-1) + 0.5\bar{r}(n) \\ \bar{q}_2(n) &= \bar{r}(n) \\ \bar{q}_3(n) &= 0.5\bar{r}(n) + 0.5\bar{q}_4(n)\end{aligned}$$

The impulse response,  $h(n)$ , of the weighted synthesis filter

$H(z)W(z) = A(z/\gamma_1)\bar{l}(z)A(z/\gamma_2)$  is computed each subframe. This impulse response is needed for the search of adaptive and fixed codebooks 257 and 261. The impulse response  $h(n)$  is computed by filtering the vector of coefficients of the filter  $A(z/\gamma_1)$  extended by zeros through the two filters  $1/\bar{A}(z)$  and  $1/A(z/\gamma_2)$ .

The target signal for the search of the adaptive codebook 257 is usually computed by subtracting the zero input response of the weighted synthesis filter  $H(z)W(z)$  from the weighted speech signal  $s_e(n)$ . This operation is performed on a frame basis. An equivalent procedure for computing the target signal is the filtering of the LP residual signal  $r(n)$  through the combination of the synthesis filter  $1/\bar{A}(z)$  and the weighting filter  $W(z)$ .

After determining the excitation for the subframe, the initial states of these filters are updated by filtering the difference between the LP residual and the excitation. The LP residual is given by:

$$r(n) = s(n) + \sum_{i=1}^{10} \bar{a}_i s(n-i), n = 0, L\_SF - 1$$

The residual signal  $r(n)$  which is needed for finding the target vector is also used in the adaptive codebook search to extend the past excitation buffer. This simplifies the adaptive codebook search procedure for delays less than the subframe size of 40 samples.

In the present embodiment, there are two ways to produce an LTP contribution. One uses pitch preprocessing (PP) when the PP-mode is selected, and another is computed like the traditional LTP when the LTP-mode is chosen. With the PP-mode, there is no need to do the adaptive codebook search, and LTP excitation is directly computed according to past synthesized excitation because the interpolated pitch contour is set for each frame. When the AMR coder operates with LTP-mode, the pitch lag is constant within one subframe, and searched and coded on a subframe basis.

Suppose the past synthesized excitation is memorized in  $(\text{ext}(MAX\_LAG+n), n < 0)$ , which is also called adaptive codebook. The LTP excitation codevector, temporally memorized in  $(\text{ext}(MAX\_LAG+n), 0 < n < L\_SF)$ , is calculated by interpolating the past excitation (adaptive

codebook) with the pitch lag contour,  $\tau_c(n+m \cdot L\_SF)$ ,  $m = 0, 1, 2, 3$ . The interpolation is performed using an FIR filter (Hamming windowed sinc functions):

$$\text{ext}(MAX\_LAG+n) = \sum_{m=0}^3 \text{ext}(MAX\_LAG+n - T_c(n) + 1) \cdot I_m(I_c(n)), n = 0, 1, \dots, L\_SF - 1;$$

where  $T_c(n)$  and  $I_c(n)$  are calculated by

$$T_c(n) = \text{round}(\tau_c(n + m \cdot L\_SF)),$$

$$I_c(n) = \tau_c(n) - T_c(n),$$

$m$  is subframe number,  $(I_c(n))$  is a set of interpolation coefficients,  $f_j$  is 10,  $MAX\_LAG$  is 145+11, and  $L\_SF=40$  is the subframe size. Note that the interpolated values

$(\text{ext}(MAX\_LAG+n), 0 < n < L\_SF - 17 + 11)$  might be used again to do the interpolation when the pitch lag is small. Once the interpolation is finished, the adaptive codevector  $V_a = (v_d(n), n=0$  to 39) is obtained by copying the interpolated values:

$$v_d(n) = \text{ext}(MAX\_LAG+n), 0 < n < L\_SF$$

Adaptive codebook searching is performed on a subframe basis. It consists of performing closed-loop pitch lag search, and then computing the adaptive code vector by interpolating the past excitation at the selected fractional pitch lag. The LTP parameters (or the adaptive codebook parameters) are the pitch lag (or the delay) and gain of the pitch filter. In the search stage, the excitation is extended by the LP residual to simplify the closed-loop search.

For the bit rate of 11.0 kbps, the pitch delay is encoded with 9 bits for the 1<sup>st</sup> and 3<sup>rd</sup> subframes and the relative delay of the other subframes is encoded with 6 bits. A fractional pitch delay is used in the first and third subframes with resolutions: 1/6 in the range  $[17, 93 \frac{1}{6}]$ , and

integers only in the range [95, 145]. For the second and fourth subframes, a pitch resolution of

1/6 is always used for the rate 11.0 kbps in the range  $[T_1 - 5\frac{3}{6}, T_1 + 4\frac{3}{6}]$ , where  $T_1$  is the pitch lag of the previous ( $1^{st}$  or  $3^{rd}$ ) subframe.

The close-loop pitch search is performed by minimizing the mean-square weighted error between the original and synthesized speech. This is achieved by maximizing the term:

$$R(k) = \frac{\sum_{n=0}^M T_p(n) y_t(n)}{\sqrt{\sum_{n=0}^M y_t(n) y_t(n)}}, \text{ where } T_p(n) \text{ is the target signal and } y_t(n) \text{ is the past filtered}$$

excitation at delay  $k$  (past excitation convoluted with  $h(n)$ ). The convolution  $y_t(n)$  is

computed for the first delay  $l_{\min}$  in the search range, and for the other delays in the search range

$k = l_{\min} + 1, \dots, l_{\max}$ , it is updated using the recursive relation:

$$y_t(n) = y_{t-1}(n-1) + u(-)h(n),$$

where  $u(n), n = -(143+1)$  to 39 is the excitation buffer.

Note that in the search stage, the samples  $u(n), n = 0$  to 39, are not available and are needed for pitch delays less than 40. To simplify the search, the LP residual is copied to  $u(n)$  to make the relation in the calculations valid for all delays. Once the optimum integer pitch delay is determined, the fractions, as defined above, around that integer are tested. The fractional pitch search is performed by interpolating the normalized correlation and searching for its maximum.

Once the fractional pitch lag is determined, the adaptive codebook vector,  $v(n)$ , is computed by interpolating the past excitation  $u(n)$  at the given phase (fraction). The interpolations are performed using two FIR filters (Hamming windowed sinc functions), one for interpolating the term in the calculations to find the fractional pitch lag and the other for

interpolating the past excitation as previously described. The adaptive codebook gain,  $g_p$ , is

temporally given then by:

$$g_p = \frac{\sum_{n=0}^M T_p(n) y(n)}{\sum_{n=0}^M y(n) y(n)},$$

bounded by  $0 < g_p < 1.2$ , where  $y(n) = v(n) * h(n)$  is the filtered adaptive

codebook vector (zero state response of  $H(z)W(z)$  to  $v(n)$ ). The adaptive codebook gain could be modified again due to joint optimization of the gains, gain normalization and smoothing. The term  $y(n)$  is also referred to herein as  $C_p(n)$ .

With conventional approaches, pitch lag maximizing correlation might result in two or more times the correct one. Thus, with such conventional approaches, the candidate of shorter pitch lag is favored by weighting the correlations of different candidates with constant weighting coefficients. At times this approach does not correct the double or treble pitch lag because the weighting coefficients are not aggressive enough or could result in halving the pitch lag due to the strong weighting coefficients.

In the present embodiment, these weighting coefficients become adaptive by checking if the present candidate is in the neighborhood of the previous pitch lags (when the previous frames are voiced) and if the candidate of shorter lag is in the neighborhood of the value obtained by dividing the longer lag (which maximizes the correlation) with an integer.

In order to improve the perceptual quality, a speech classifier is used to direct the searching procedure of the fixed codebook (as indicated by the blocks 275 and 279) and to control gain normalization (as indicated in the block 401 of Fig. 4). The speech classifier serves to improve the background noise performance for the lower rate coders, and to get a quick start-

up of the noise level estimation. The speech classifier distinguishes stationary noise-like segments from segments of speech, music, tonal-like signals, non-stationary noise, etc.

The speech classification is performed in two steps. An initial classification

(*speech\_mode*) is obtained based on the modified input signal. The final classification (*exc\_mode*) is obtained from the initial classification and the residual signal after the pitch contribution has been removed. The two outputs from the speech classification are the excitation mode, *exc\_mode*, and the parameter  $\beta_{ms}(n)$ , used to control the subframe based smoothing of the gains.

The speech classification is used to direct the encoder according to the characteristics of the input signal and need not be transmitted to the decoder. Thus, the bit allocation, codebooks, and decoding remain the same regardless of the classification. The encoder emphasizes the perceptually important features of the input signal on a subframe basis by adapting the encoding in response to such features. It is important to notice that misclassification will not result in disastrous speech quality degradations. Thus, as opposed to the VAD 235, the speech classifier identified within the block 279 (Fig. 2) is designed to be somewhat more aggressive for optimal perceptual quality.

The initial classifier (*speech\_classifier*) has adaptive thresholds and is performed in six steps:

1. Adapt thresholds:
  - if (*updates\_noise*  $\geq 30$  & *updates\_speech*  $\geq 30$ )
    - $SNR_{max} = \min \left( \frac{ma\_max\_speech}{ma\_max\_noise}, 32 \right)$
    - else
    - $SNR_{max} = 3.5$
    - endif
  - if ( $SNR_{max} < 1.75$ )
    - $dec\_max\_mes = 1.30$
    - $dec\_ma\_cp = 0.70$
    - $update\_max\_mes = 1.10$
    - $update\_ma\_cp\_speech = 0.72$
    - elseif ( $SNR_{max} < 2.50$ )
      - $dec\_max\_mes = 1.65$
      - $dec\_ma\_cp = 0.73$
      - $update\_max\_mes = 1.30$
      - $update\_ma\_cp\_speech = 0.72$
    - else
    - $dec\_max\_mes = 1.75$
    - $dec\_ma\_cp = 0.77$
    - $update\_max\_mes = 1.30$
    - $update\_ma\_cp\_speech = 0.77$
    - endif
  - 2. Calculate parameters:

Pitch correlation:

$$cp = \frac{\sum_{i=0}^{L-1} \tilde{s}(i) \cdot \tilde{s}(i - lag)}{\sqrt{\left( \sum_{i=0}^{L-1} \tilde{s}(i) \cdot \tilde{s}(i) \right) \cdot \left( \sum_{i=0}^{L-1} \tilde{s}(i - lag) \cdot \tilde{s}(i - lag) \right)}}$$

Running mean of pitch correlation:

$$ma\_cp(n) = 0.9 \cdot ma\_cp(n-1) + 0.1 \cdot cp$$

Maximum of signal amplitude in current pitch cycle:

$$max(n) = \max\{s(i) | i = start, \dots, L\_SF - 1\}$$

where:

$$start = \min(L\_SF - lag, 0)$$

Sum of signal amplitudes in current pitch cycle:

$$mean(n) = \sum_{i=start}^{L\_SF-1} |s(i)|$$

Measure of relative maximum:

$$ma\_mes = \frac{max(n)}{ma\_max\_noise(n-1)}$$

Maximum to long-term sum:

$$max2sum = \frac{max(n)}{\sum_{k=1}^n mean(n-k)}$$

Maximum in groups of 3 subframes for past 15 subframes:

$$max\_group(n, k) = \max\{max(n-3 \cdot (4-k) - j) | j = 0, \dots, 2\} \quad k = 0, \dots, 4$$

Group-maximum to minimum of previous 4 group-maxima:

$$endmax2minmax = \frac{max\_group(n, 4)}{\min\{max\_group(n, k) | k = 0, \dots, 3\}}$$

Slope of 5 group maxima:

$$slope = 0.1 \cdot \sum_{k=0}^4 (k-2) \cdot max\_group(n, k)$$

3. Classify subframe:

```

if (((max_mes < deci_max_mes & ma_cp < deci_ma_cp) || (VAD == 0)) &
    (LTP_MODE == 115.8kbit/s || 14.55kbit/s))
    speech_mode = 0 /* class1 */
else
    speech_mode = 1 /* class2 */
endif

```

4. Check for change in background noise level, i.e. reset required:

Check for decrease in level:

```

if (updates_noise == 31 & max_mes <= 0.3)
    if (consec_low < 15)
        consec_low++
    endif
else
    consec_low = 0
endif

```

```

if (consec_low == 15)
    updates_noise = 0
    lev_reset = -1 /* low level reset */
endif

```

Check for increase in level:

```

if ((updates_noise >= 30 || lev_reset == -1) & max_mes > 1.5 & ma_cp < 0.70 & cp < 0.85
    & k1 < -0.4 & endmax2minmax < 50 & max2sum < 35 & slope > -100 & slope < 120)
    if (consec_high < 15)
        consec_high++
    endif
else
    consec_high = 0
endif

```

```

if (consec_high == 15 & endmax2minmax < 6 & max2sum < 5)
    updates_noise = 30
    lev_reset = 1 /* high level reset */
endif

```

5. Update running mean of maximum of class 1 segments, i.e. stationary noise:

```

if(
/*1. condition: regular update */
(max_mes < update_max_mes & ma_cp < 0.6 & cp < 0.65 & max_mes > 0.3) |
/*2. condition: VAD continued update */
(consec_vad_0 = 8) |
/*3. condition: start - up/reset update */
(update_noise ≤ 30 & ma_cp < 0.7 & cp < 0.75 & k1 < -0.4 & endmax2minimax < 5 &
(lev_reset ≠ -1) (lev_reset = -1 & max_mes < 2)))
)
ma_max_noise(n) = 0.9 · ma_max_noise(n-1) + 0.1 · max(n)

```

```

if (update_noise ≤ 30)
updates_noise ++
else
lev_reset = 0
endif

```

where  $k_1$  is the first reflection coefficient.

6. Update running mean of maximum of class 2 segments, i.e. speech, music, tonal-like signal, non-stationary noise, etc. continued from above:

```

elseif (ma_cp > update_ma_cp_speech)
if (update_speech ≤ 80)
αspeech = 0.95
else
αspeech = 0.999
endif

```

$ma\_max\_speech(n) = \alpha_{speech} \cdot ma\_max\_speech(n-1) + (1 - \alpha_{speech}) \cdot max(n)$

```

if (update_speech ≤ 80)
updates_speech ++
endif

```

-47-

The final classifier (*exc\_preselct*) provides the final class, *exc\_mode*, and the subframe based smoothing parameter,  $\beta_{ms}(n)$ . It has three steps:

1. Calculate parameters:

Maximum amplitude of ideal excitation in current subframe:

$$max_{ms}(n) = \max\{res2(i), i = 0, \dots, L_{SF} - 1\}$$

Measure of relative maximum:

$$max\_mes_{ms} = \frac{max_{ms}(n)}{ma\_max_{ms}(n-1)}$$

2. Classify subframe and calculate smoothing:

if (*speech\_mode* = 1 | *max\_mes\_ms* ≥ 1.75)

*exc\_mode* = 1 /\*class 2 \*/

$\beta_{ms}(n) = 0$

$N\_mode\_sub(n) = -4$

else

*exc\_mode* = 0 /\*class 1 \*/

$N\_mode\_sub(n) = N\_mode\_sub(n-1) + 1$

if ( $N\_mode\_sub(n) > 4$ )

$N\_mode\_sub(n) = 4$

endif

if ( $N\_mode\_sub(n) > 0$ )

$$\beta_{ms}(n) = \frac{0.7}{9} \cdot (N\_mode\_sub(n) - 1)^2$$

else

$\beta_{ms}(n) = 0$

endif

endif

-48-



3. Update running mean of maximum:

```

if (max_mes_max ≤ 0.5)
  if (consec < 51)
    consec ++
  endif
else
  consec = 0
endif
if ((exc_mode = 0 & (max_mes_max > 0.51) & consec > 50)) |
  (updates ≤ 30 & ma_cp < 0.6 & cp < 0.65))
  ma_max(n) = 0.9 · ma_max(n-1) + 0.1 · max_mes(n)
  if (updates ≤ 30)
    updates ++
  endif
endif

```

When this process is completed, the final subframe based classification, exc\_mode, and the smoothing parameter,  $\beta_{sub}(n)$ , are available.

To enhance the quality of the search of the fixed codebook 261, the target signal,  $T_t(n)$ , is produced by temporally reducing the LTP contribution with a gain factor,  $G_t$ :

$$T_t(n) = T_{st}(n) \cdot G_t \cdot g_p \cdot Y_s(n), \quad n=0,1,\dots,39$$

where  $T_{st}(n)$  is the original target signal 253,  $Y_s(n)$  is the filtered signal from the adaptive codebook,  $g_p$  is the LTP gain for the selected adaptive codebook vector, and the gain factor is determined according to the normalized LTP gain,  $R_p$ , and the bit rate:

if (rate ≤ 0) /\* for 4.5kbps and 5.8kbps \*/  
 $G_t = 0.7 R_p + 0.3$ ;

if (rate = 1) /\* for 6.65kbps \*/  
 $G_t = 0.6 R_p + 0.4$ ;

if (rate == 2) /\* for 8.0kbps \*/  
 $G_t = 0.3 R_p + 0.7$ ;

if (rate == 3) /\* for 11.0kbps \*/  
 $G_t = 0.95$ ;

if ( $T_{op} > L_{SF}$  &  $g_p > 0.5$  & rate ≤ 2)  
 $G_t = G_t \cdot (0.3 \cdot R_p + 0.7)$ ; and

where normalized LTP gain,  $R_p$ , is defined as:

$$R_p = \frac{\sum_{n=0}^{39} T_{st}(n) Y_s(n)}{\sqrt{\sum_{n=0}^{39} T_{st}(n) T_{st}(n)} \cdot \sqrt{\sum_{n=0}^{39} Y_s(n) Y_s(n)}}$$

Another factor considered at the control block 275 in conducting the fixed codebook search and at the block 401 (Fig. 4) during gain normalization is the noise level + "γ" which is given by:

$$P_{Ksz} = \sqrt{\frac{\max((E_n - 100), 0.0)}{E_n}}$$

where  $E_n$  is the energy of the current input signal including background noise, and  $E_n$  is a running average energy of the background noise.  $E_n$  is updated only when the input signal is detected to be background noise as follows:

if (first background noise frame is true)  
 $E_n = 0.75 E_i$ ;  
 else if (background noise frame is true)  
 $E_n = 0.75 E_{n-1} + 0.25 E_i$ ;

where  $E_{n-1}$  is the last estimation of the background noise energy.

For each bit rate mode, the fixed codebook 261 (Fig. 2) consists of two or more subcodebooks which are constructed with different structure. For example, in the present embodiment at higher rates, all the subcodebooks only contain pulses. At lower bit rates, one of

the subcodebooks is populated with Gaussian noise. For the lower bit-rates (e.g., 6.65, 5.8, 4.55 kbps), the speech classifier forces the encoder to choose from the Gaussian subcodebook in case of stationary noise-like subframes, *exc\_mode* = 0. For *exc\_mode* = 1 all subcodebooks are searched using adaptive weighting.

For the pulse subcodebooks, a fast searching approach is used to choose a subcodebook and select the code word for the current subframe. The same searching routine is used for all the bit rate modes with different input parameters.

In particular, the long-term enhancement filter,  $F_p(z)$ , is used to filter through the selected pulse excitation. The filter is defined as  $F_p(z) = \frac{1}{1 - \beta z^{-T}}$ , where  $T$  is the integer part of

pitch lag at the center of the current subframe, and  $\beta$  is the pitch gain of previous subframe, bounded by [0.2, 1.0]. Prior to the codebook search, the impulsive response  $h(n)$  includes the filter  $F_p(z)$ .

For the Gaussian subcodebooks, a special structure is used in order to bring down the storage requirement and the computational complexity. Furthermore, no pitch enhancement is applied to the Gaussian subcodebooks.

There are two kinds of pulse subcodebooks in the present AMR coder embodiment. All pulses have the amplitudes of +1 or -1. Each pulse has 0, 1, 2, 3 or 4 bits to code the pulse position. The signs of some pulses are transmitted to the decoder with one bit coding one sign. The signs of other pulses are determined in a way related to the coded signs and their pulse positions.

In the first kind of pulse subcodebook, each pulse has 3 or 4 bits to code the pulse position. The possible locations of individual pulses are defined by two basic non-regular tracks and initial phases:

$$POS(n_p, i) = TRACK(m_p, i) + PHAS(n_p, phase\_mode),$$

where  $i=0, 1, \dots, 7$  or 15 (corresponding to 3 or 4 bits to code the position), is the possible position index,  $n_p = 0, \dots, N_p-1$  ( $N_p$  is the total number of pulses), distinguishes different pulses,  $m_p=0$  or 1, defines two tracks, and *phase\_mode*=0 or 1, specifies two phase modes.

For 3 bits to code the pulse position, the two basic tracks are:

$$\begin{aligned} TRACK(0,i) &= (0, 4, 8, 12, 18, 24, 30, 36), \text{ and} \\ TRACK(1,i) &= (0, 6, 12, 18, 22, 26, 30, 34). \end{aligned}$$

If the position of each pulse is coded with 4 bits, the basic tracks are:

$$\begin{aligned} TRACK(0,i) &= (0, 2, 4, 6, 8, 10, 12, 14, 17, 20, 23, 26, 29, 32, 35, 38), \text{ and} \\ TRACK(1,i) &= (0, 3, 6, 9, 12, 15, 18, 21, 23, 25, 27, 29, 31, 33, 35, 37). \end{aligned}$$

The initial phase of each pulse is fixed as:

$$\begin{aligned} PHAS(n_p, 0) &= \text{modulus}(n_p / MAXPHAS) \\ PHAS(n_p, 1) &= PHAS(N_p - 1 - n_p, 0) \end{aligned}$$

where *MAXPHAS* is the maximum phase value.

For any pulse subcodebook, at least the first sign for the first pulse,  $SIGN(n_p)$ ,  $n_p=0$ , is encoded because the gain sign is embedded. Suppose  $N_{gain}$  is the number of pulses with encoded signs, that is,  $SIGN(n_p)$ , for  $n_p < N_{gain}$ ,  $n_p \leq N_p$ , is encoded while  $SIGN(n_p)$ , for  $n_p > N_{gain}$ , is not encoded. Generally, all the signs can be determined in the following way:

$$SIGN(n_p) = -SIGN(n_p - 1), \text{ for } n_p > N_{gain},$$

due to that the pulse positions are sequentially searched from  $n_p=0$  to  $n_p=N_p-1$  using an iteration approach. If two pulses are located in the same track while only the sign of the first pulse in the track is encoded, the sign of the second pulse depends on its position relative to the first pulse. If the position of the second pulse is smaller, then it has opposite sign, otherwise it has the same sign as the first pulse.

In the second kind of pulse subcodebook, the innovation vector contains 10 signed pulses. Each pulse has 0, 1, or 2 bits to code the pulse position. One subframe with the size of 40 samples is divided into 10 small segments with the length of 4 samples. 10 pulses are respectively located into 10 segments. Since the position of each pulse is limited into one segment, the possible locations for the pulse numbered with  $n_p$  are,  $(4n_p)$ ,  $(4n_p + 2)$ , or  $(4n_p + 4n_p + 1)$ ,  $(4n_p + 2)$ ,  $(4n_p + 3)$ , respectively for 0, 1, or 2 bits to code the pulse position. All the signs for all the 10 pulses are encoded.

The fixed codebook 261 is searched by minimizing the mean square error between the weighted input speech and the weighted synthesized speech. The target signal used for the LTP excitation is updated by subtracting the adaptive codebook contribution. That is:

$$x_2(n) = x(n) - \hat{g}_p y(n), \quad n = 0, \dots, 39,$$

where  $y(n) = v(n) * h(n)$  is the filtered adaptive codebook vector and  $\hat{g}_p$  is the modified (reduced) LTP gain.

If  $c_k$  is the code vector at index  $k$  from the fixed codebook, then the pulse codebook is searched by maximizing the term:

$$A_k = \frac{(c_k)^T \cdot (d^T c_k)}{E_{Dk} \cdot c_k^T \Phi c_k},$$

where  $d = H^T x_1$  is the correlation between the target signal  $x_1(n)$  and the impulse response  $h(n)$ ,  $H$  is the lower triangular Toeplitz convolution matrix with diagonal  $h(0)$  and lower diagonals  $h(1), \dots, h(39)$ , and  $\Phi = H^T H$  is the matrix of correlations of  $h(n)$ . The vector  $d$  (backward filtered target) and the matrix  $\Phi$  are computed prior to the codebook search. The elements of the vector  $d$  are computed by:

-53-

$$d(n) = \sum_{i=n}^{39} x_2(i) h(i-n), \quad n = 0, \dots, 39,$$

and the elements of the symmetric matrix  $\Phi$  are computed by:

$$\phi(i, j) = \sum_{n=j}^{39} h(n-i) h(n-j), \quad (j \geq i),$$

The correlation in the numerator is given by:

$$C = \sum_{i=0}^{N_p-1} \phi_i d(m_i),$$

where  $m_i$  is the position of the  $i$ th pulse and  $\phi_i$  is its amplitude. For the complexity reason, all the amplitudes  $\{\phi_i\}$  are set to +1 or -1; that is,

$$\phi_i = \text{SIGN}(i), \quad i = n_p = 0, \dots, N_p - 1.$$

The energy in the denominator is given by:

$$E_D = \sum_{i=0}^{N_p-1} \phi_i(m_i, m_i) + 2 \sum_{i=0}^{N_p-2} \sum_{j=i+1}^{N_p-1} \phi_i \phi_j \phi(m_i, m_j).$$

To simplify the search procedure, the pulse signs are preset by using the signal  $x(n)$ ,

which is a weighted sum of the normalized  $d(n)$  vector and the normalized target signal of  $x_2(n)$  in the residual domain  $res_2(n)$ :

$$b(n) = \frac{res_2(n)}{\sqrt{\sum_{i=0}^{39} res_2(i)^2}} + \frac{2d(n)}{\sqrt{\sum_{i=0}^{39} d(i)^2}}, \quad n = 0, 1, \dots, 39$$

If the sign of the  $i$ th ( $i = n_p$ ) pulse located at  $m_i$  is encoded, it is set to the sign of signal  $b(n)$  at that position, i.e.,  $\text{SIGN}(i) = \text{sign}[b(m_i)]$ .

-54-

In the present embodiment, the fixed codebook 261 has 2 or 3 subcodebooks for each of the encoding bit rates. Of course many more might be used in other embodiments. Even with several subcodebooks, however, the searching of the fixed codebook 261 is very fast using the following procedure. In a first searching turn, the encoder processing circuitry searches the pulse positions sequentially from the first pulse ( $n_p=0$ ) to the last pulse ( $n_p=N_p-1$ ) by considering the influence of all the existing pulses.

In a second searching turn, the encoder processing circuitry corrects each pulse position sequentially from the first pulse to the last pulse by checking the criterion value  $A_i$  contributed from all the pulses for all possible locations of the current pulse. In a third turn, the functionality of the second searching turn is repeated a final time. Of course further turns may be utilized if the added complexity is not prohibitive.

The above searching approach proves very efficient, because only one position of one pulse is changed leading to changes in only one term in the criterion numerator  $C$  and few terms in the criterion denominator  $E_0$  for each computation of the  $A_i$ . As an example, suppose a pulse subcodebook is constructed with 4 pulses and 3 bits per pulse to encode the position. Only 96 ( $4 \text{ pulses} \times 2^3 \text{ positions per pulse} \times 3 \text{ turns} = 96$ ) simplified computations of the criterion  $A_i$  need be performed.

Moreover, to save the complexity, usually only one of the subcodebooks in the fixed codebook 261 is chosen after finishing the first searching turn. Further searching turns are done only with the chosen subcodebook. In other embodiments, one of the subcodebooks might be chosen only after the second searching turn or thereafter should processing resources so permit.

The Gaussian codebook is structured to reduce the storage requirement and the computational complexity. A comb-structure with two basis vectors is used. In the comb-

structure, the basis vectors are orthogonal, facilitating a low complexity search. In the AMR coder, the first basis vector occupies the even sample positions, (0,2,...,38), and the second basis vector occupies the odd sample positions, (1,3,...,39).

The same codebook is used for both basis vectors, and the length of the codebook vectors is 20 samples (half the subframe size).

All rates (6.65, 5.8 and 4.55 kbps) use the same Gaussian codebook. The Gaussian codebook,  $CB_{\text{Gauss}}$ , has only 10 entries, and thus the storage requirement is  $10 \cdot 20 = 200$  16-bit words. From the 10 entries, as many as 32 code vectors are generated. An index,  $idx_i$ , to one basis vector 22 populates the corresponding part of a code vector,  $c_{w_i}$ , in the following way:

$$c_{w_i}(2 \cdot (i - \tau) + \delta) = CB_{\text{Gauss}}(l, i) \quad i = \tau, \tau + 1, \dots, 19$$

$$c_{w_i}(2 \cdot (i + 20 - \tau) + \delta) = CB_{\text{Gauss}}(l, i) \quad i = 0, 1, \dots, \tau - 1$$

where the table entry,  $l$ , and the shift,  $\tau$ , are calculated from the index,  $idx_i$ , according to:

$$\tau = \text{trunc}(idx_i / 10)$$

$$l = idx_i - 10 \cdot \tau$$

and  $\delta$  is 0 for the first basis vector and 1 for the second basis vector. In addition, a sign is applied to each basis vector.

Basically, each entry in the Gaussian table can produce as many as 20 unique vectors, all with the same energy due to the circular shift. The 10 entries are all normalized to have identical energy of 0.5, i.e.,

$$\sum_{i=0}^{19} CB_{\text{Gauss}}(l, i)^2 = 0.5, \quad l = 0, 1, \dots, 9$$

This means that when both basis vectors have been selected, the combined code vector,  $c_{w_0, w_1}$ , will have unity energy, and thus the final excitation vector from the Gaussian subcodebook will

have unity energy since no pitch enhancement is applied to candidate vectors from the Gaussian subcodebook.

The search of the Gaussian codebook utilizes the structure of the codebook to facilitate a low complexity search. Initially, the candidates for the two basis vectors are searched independently based on the ideal excitation,  $res_1$ . For each basis vector, the two best candidates, along with the respective signs, are found according to the mean squared error. This is exemplified by the equations to find the best candidate, index  $idx_1$ , and its sign,  $s_{idx_1}$ :

$$idx_1 = \max_{i=1, \dots, N_{Gauss}} \left\{ \sum_{n=0}^{10} res_1(2 \cdot i + \delta) \cdot c_1(2 \cdot i + \delta) \right\}$$

$$s_{idx_1} = \text{sign} \left( \sum_{n=0}^{10} res_1(2 \cdot i + \delta) \cdot c_{idx_1}(2 \cdot i + \delta) \right)$$

where  $N_{Gauss}$  is the number of candidate entries for the basis vector. The remaining parameters are explained above. The total number of entries in the Gaussian codebook is  $2 \cdot 2 \cdot N_{Gauss}^2$ . The fine search minimizes the error between the weighted speech and the weighted synthesized speech considering the possible combination of candidates for the two basis vectors from the pre-selection. If  $c_{k_0, k_1}$  is the Gaussian code vector from the candidate vectors represented by the indices  $k_0$  and  $k_1$  and the respective signs for the two basis vectors, then the final Gaussian code vector is selected by maximizing the term:

$$A_{k_0, k_1} = \frac{(c_{k_0, k_1})^T (d^T c_{k_0, k_1})}{E_{d, k_0, k_1} \cdot c_{k_0, k_1}^T \Phi c_{k_0, k_1}}$$

over the candidate vectors.  $d = H^T x_1$  is the correlation between the target signal  $x_1(n)$  and the impulse response  $h(n)$  (without the pitch enhancement), and  $H$  is a the lower triangular Toeplitz

convolution matrix with diagonal  $h(0)$  and lower diagonals  $h(1), \dots, h(39)$ , and  $\Phi = H^T H$  is the matrix of correlations of  $h(n)$ .

More particularly, in the present embodiment, two subcodebooks are included (or utilized) in the fixed codebook 261 with 31 bits in the 11 kbps encoding mode. In the first subcodebook, the innovation vector contains 8 pulses. Each pulse has 3 bits to code the pulse position. The signs of 6 pulses are transmitted to the decoder with 6 bits. The second subcodebook contains innovation vectors comprising 10 pulses. Two bits for each pulse are assigned to code the pulse position which is limited in one of the 10 segments. Ten bits are spent for 10 signs of the 10 pulses. The bit allocation for the subcodebooks used in the fixed codebook 261 can be summarized as follows:

*Subcodebook1: 8 pulses X 3 bits/pulse + 6 signs = 30 bits*

*Subcodebook2: 10 pulses X 2 bits/pulse + 10 signs = 30 bits*

One of the two subcodebooks is chosen at the block 275 (Fig. 2) by favoring the second subcodebook using adaptive weighting applied when comparing the criterion value  $F1$  from the first subcodebook to the criterion value  $F2$  from the second subcodebook:

*if ( $W_c \cdot F1 > F2$ ), the first subcodebook is chosen,*  
*else, the second subcodebook is chosen,*

where the weighting,  $0 < W_c \leq 1$ , is defined as:

$$W_c = \begin{cases} 1.0, & \text{if } P_{KSR} < 0.5, \\ 1.0 - 0.3 P_{KSR} (1.0 - 0.5 R_p) \cdot \min(P_{sharp} + 0.5, 1.0), & \text{else} \end{cases}$$

$P_{KSR}$  is the background noise to speech signal ratio (i.e., the "noise level" in the block 279),  $R_p$  is the normalized LTP gain, and  $P_{sharp}$  is the sharpness parameter of the ideal excitation  $res(n)$  (i.e., the "sharpness" in the block 279).

In the 8 kbps mode, two subcodebooks are included in the fixed codebook 261 with 20

bits. In the first subcodebook, the innovation vector contains 4 pulses. Each pulse has 4 bits to code the pulse position. The signs of 3 pulses are transmitted to the decoder with 3 bits. The second subcodebook contains innovation vectors having 10 pulses. One bit for each of 9 pulses is assigned to code the pulse position which is limited in one of the 10 segments. Ten bits are spent for 10 signs of the 10 pulses. The bit allocation for the subcodebook can be summarized as the following:

*Subcodebook1: 4 pulses X 4 bits/pulse + 3 signs =19 bits*  
*Subcodebook2: 9 pulses X 1 bits/pulse + 1 pulse X 0 bit + 10 signs =19 bits*

One of the two subcodebooks is chosen by favoring the second subcodebook using adaptive weighting applied when comparing the criterion value  $F_1$  from the first subcodebook to the criterion value  $F_2$  from the second subcodebook as in the 11 kbps mode. The weighting,  $0 < W_c \leq 1$ , is defined as:

$$W_c = 10 - 0.6 P_{\text{NSR}} (1.0 - 0.5 R_p) \cdot \min(P_{\text{NSR}} + 0.5, 1.0).$$

The 6.65 kbps mode operates using the long-term preprocessing (PP) or the traditional LTP. A pulse subcodebook of 18 bits is used when in the PP-mode. A total of 13 bits are allocated for three subcodebooks when operating in the LTP-mode. The bit allocation for the subcodebooks can be summarized as follows:

*PP-mode:*  
*Subcodebook: 5 pulses X 3 bits/pulse + 3 signs =18 bits*  
*LTP-mode:*  
*Subcodebook1: 3 pulses X 3 bits/pulse + 3 signs =12 bits, phase\_mode=1,*  
*Subcodebook2: 3 pulses X 3 bits/pulse + 2 signs =11 bits, phase\_mode=0,*  
*Subcodebook3: Gaussian subcodebook of 11 bits.*

One of the 3 subcodebooks is chosen by favoring the Gaussian subcodebook when searching with LTP-mode. Adaptive weighting is applied when comparing the criterion value from the

two pulse subcodebooks to the criterion value from the Gaussian subcodebook. The weighting,

$0 < W_c \leq 1$ , is defined as:

$$W_c = 10 - 0.9 P_{\text{NSR}} (1.0 - 0.5 R_p) \cdot \min(P_{\text{NSR}} + 0.5, 1.0),$$

$$\text{if (noise - like unvoiced), } W_c \leftarrow W_c \cdot (0.2 R_p (1.0 - P_{\text{NSR}}) + 0.8).$$

The 5.8 kbps encoding mode works only with the long-term preprocessing (PP). Total 14 bits are allocated for three subcodebooks. The bit allocation for the subcodebooks can be summarized as the following:

*Subcodebook1: 4 pulses X 3 bits/pulse + 1 signs =13 bits, phase\_mode=1,*  
*Subcodebook2: 3 pulses X 3 bits/pulse + 3 signs =12 bits, phase\_mode=0,*  
*Subcodebook3: Gaussian subcodebook of 12 bits.*

One of the 3 subcodebooks is chosen favoring the Gaussian subcodebook with adaptive weighting applied when comparing the criterion value from the two pulse subcodebooks to the criterion value from the Gaussian subcodebook. The weighting,  $0 < W_c \leq 1$ , is defined as:

$$W_c = 1.0 - P_{\text{NSR}} (1.0 - 0.5 R_p) \cdot \min(P_{\text{NSR}} + 0.6, 1.0),$$

$$\text{if (noise - like unvoiced), } W_c \leftarrow W_c \cdot (0.3 R_p (1.0 - P_{\text{NSR}}) + 0.7).$$

The 4.55 kbps bit rate mode works only with the long-term preprocessing (PP). Total 10 bits are allocated for three subcodebooks. The bit allocation for the subcodebooks can be summarized as the following:

*Subcodebook1: 2 pulses X 4 bits/pulse + 1 signs =9 bits, phase\_mode=1,*  
*Subcodebook2: 2 pulses X 3 bits/pulse + 2 signs =8 bits, phase\_mode=0,*  
*Subcodebook3: Gaussian subcodebook of 8 bits.*

One of the 3 subcodebooks is chosen by favoring the Gaussian subcodebook with weighting applied when comparing the criterion value from the two pulse subcodebooks to the criterion value from the Gaussian subcodebook. The weighting,  $0 < W_c \leq 1$ , is defined as:

$$W_c = 10 - 1.2 P_{\text{NSR}} (1.0 - 0.5 R_p) \cdot \min(P_{\text{NSR}} + 0.6, 1.0).$$

if (noise - like unvoiced),  $W_c \Leftarrow W_c \cdot (0.6 R_p (1.0 - P_{loop}) + 0.4)$ .

For 4.55, 5.8, 6.65 and 8.0 kbps bit rate encoding modes, a gain re-optimization

procedure is performed to jointly optimize the adaptive and fixed codebook gains,  $g_p$  and  $g_c$ ,

respectively, as indicated in Fig. 3. The optimal gains are obtained from the following

correlations given by:

$$g_p = \frac{R_1 R_2 - R_3 R_4}{R_2 R_3 - R_1 R_4},$$

$$g_c = \frac{R_4 - g_p R_3}{R_3},$$

where  $R_1 \Leftarrow \langle \bar{C}_p, \bar{T}_p \rangle$ ,  $R_2 \Leftarrow \langle \bar{C}_c, \bar{C}_c \rangle$ ,  $R_3 \Leftarrow \langle \bar{C}_p, \bar{C}_c \rangle$ ,  $R_4 \Leftarrow \langle \bar{C}_c, \bar{T}_p \rangle$ , and

$R_5 \Leftarrow \langle \bar{C}_p, \bar{C}_p \rangle$ ,  $\bar{C}_c$ ,  $\bar{T}_p$  and  $\bar{T}_p$  are filtered fixed codebook excitation, filtered adaptive codebook excitation and the target signal for the adaptive codebook search.

For 11 kbps bit rate encoding, the adaptive codebook gain,  $g_p$ , remains the same as that computed in the close-loop pitch search. The fixed codebook gain,  $g_c$ , is obtained as:

$$g_c = \frac{R_5}{R_2},$$

where  $R_5 \Leftarrow \langle \bar{C}_p, \bar{T}_p \rangle$  and  $\bar{T}_p = \bar{T}_p - g_p \bar{C}_p$ .

Original CELP algorithm is based on the concept of analysis by synthesis (waveform matching). At low bit rate or when coding noisy speech, the waveform matching becomes difficult so that the gains are up-down, frequently resulting in unnatural sounds. To compensate for this problem, the gains obtained in the analysis by synthesis close-loop sometimes need to be modified or normalized.

There are two basic gain normalization approaches. One is called open-loop approach which normalizes the energy of the synthesized excitation to the energy of the unquantized residual signal. Another one is close-loop approach with which the normalization is done considering the perceptual weighting. The gain normalization factor is a linear combination of the one from the close-loop approach and the one from the open-loop approach; the weighting coefficients used for the combination are controlled according to the LPC gain.

The decision to do the gain normalization is made if one of the following conditions is met: (a) the bit rate is 8.0 or 6.65 kbps, and noise-like unvoiced speech is true; (b) the noise level  $P_{NXX}$  is larger than 0.5; (c) the bit rate is 6.65 kbps, and the noise level  $P_{NXX}$  is larger than 0.2; and (d) the bit rate is 5.8 or 4.45 kbps.

The residual energy,  $E_{res}$ , and the target signal energy,  $E_{Tn}$ , are defined respectively as:

$$E_{res} = \sum_{n=0}^{L_{Tn}-1} res^2(n)$$

$$E_{Tn} = \sum_{n=0}^{L_{Tn}-1} T_n^2(n)$$

Then the smoothed open-loop energy and the smoothed closed-loop energy are evaluated by:

$$\begin{aligned} & \text{if (first subframe is true)} \\ & \quad Ol\_Eg = E_{res} \\ & \text{else} \\ & \quad Ol\_Eg \Leftarrow \beta_{mb} \cdot Ol\_Eg + (1 - \beta_{mb}) E_{res} \\ & \text{if (first subframe is true)} \\ & \quad Cl\_Eg = E_{Tn} \\ & \text{else} \\ & \quad Cl\_Eg \Leftarrow \beta_{mb} \cdot Cl\_Eg + (1 - \beta_{mb}) E_{Tn} \end{aligned}$$

where  $\beta_{\text{new}}$  is the smoothing coefficient which is determined according to the classification. After having the reference energy, the open-loop gain normalization factor is calculated:

$$ol\_g = \text{MIN}(C_d \sqrt{\frac{OL\_Eg}{\sum_{n=0}^{L\_SF-1} v^2(n)}}, \frac{1.2}{g_r})$$

where  $C_d$  is 0.8 for the bit rate 11.0 kbps, for the other rates  $C_d$  is 0.7, and  $v(n)$  is the excitation:

$$v(n) = v_d(n)g_p + v_d(n)g_c, \quad n=0,1,\dots,L\_SF-1.$$

where  $g_p$  and  $g_c$  are unquantized gains. Similarly, the closed-loop gain normalization factor is:

$$cl\_g = \text{MIN}(C_d \sqrt{\frac{CL\_Eg}{\sum_{n=0}^{L\_SF-1} y^2(n)}}, \frac{1.2}{g_r})$$

where  $C_d$  is 0.9 for the bit rate 11.0 kbps, for the other rates  $C_d$  is 0.8, and  $y(n)$  is the filtered signal ( $y(n)=v(n)*h(n)$ ):

$$y(n) = y_d(n)g_p + y_d(n)g_c, \quad n=0,1,\dots,L\_SF-1.$$

The final gain normalization factor,  $g_r$ , is a combination of  $cl\_g$  and  $ol\_g$ , controlled in terms of an LPC gain parameter,  $C_{LPC}$ :

If (speech is true or the rate is 11 kbps)

$$g_r = C_{LPC} OL\_g + (1 - C_{LPC}) CL\_g$$

$$g_r = \text{MAX}(1.0, g_r)$$

$$g_r = \text{MIN}(g_r, 1 + C_{LPC})$$

If (background noise is true and the rate is smaller than 11 kbps)

$$g_r = 1.2 \text{ MIN}(CL\_g, OL\_g)$$

where  $C_{LPC}$  is defined as:

$$C_{LPC} = \text{MIN}(\text{sqrt}(E_{\text{new}}/E_{\text{ref}}), 0.8/0.8)$$

Once the gain normalization factor is determined, the unquantized gains are modified:

$$g_r \leftarrow g_r \cdot g_r$$

For 4.55, 5.8, 6.65 and 8.0 kbps bit rate encoding, the adaptive codebook gain and the fixed codebook gain are vector quantized using 6 bits for rate 4.55 kbps and 7 bits for the other rates. The gain codebook search is done by minimizing the mean squared weighted error,  $Err$ , between the original and reconstructed speech signals:

$$Err = \|\vec{F}_n - g_p \vec{C}_p - g_c \vec{C}_c\|^2$$

For rate 11.0 kbps, scalar quantization is performed to quantize both the adaptive codebook gain,  $g_p$ , using 4 bits and the fixed codebook gain,  $g_c$ , using 5 bits each.

The fixed codebook gain,  $g_c$ , is obtained by MA prediction of the energy of the scaled fixed codebook excitation in the following manner. Let  $E(n)$  be the mean removed energy of the scaled fixed codebook excitation in (dB) at subframe  $n$  be given by:

$$E(n) = 10 \log\left(\frac{1}{40} g_c^2 \sum_{i=0}^{39} c^2(i)\right) - \bar{E}$$

where  $c(i)$  is the unscaled fixed codebook excitation, and  $\bar{E} = 30$  dB is the mean energy of scaled fixed codebook excitation.

The predicted energy is given by:

$$\bar{E}(n) = \sum_{i=0}^3 b_i \hat{R}(n-i)$$

where  $[b_0, b_1, b_2, b_3] = [0.68, 0.58, 0.34, 0.19]$  are the MA prediction coefficients and  $\hat{R}(n)$  is the quantized prediction error at subframe  $n$ .



The predicted energy is used to compute a predicted fixed codebook gain  $g_e$  (by substituting  $E(n)$  by  $\bar{E}(n)$  and  $g_e$  by  $\bar{g}_e$ ). This is done as follows. First, the mean energy of the unscaled fixed codebook excitation is computed as:

$$E_i = 10 \log \left( \frac{1}{40} \sum_{l=0}^{39} c^2(l) \right),$$

and then the predicted gain  $g_e$  is obtained as:

$$g_e = 10^{(0.05(\bar{E}(n) - E_i))}$$

A correction factor between the gain,  $g_e$ , and the estimated one,  $\bar{g}_e$ , is given by:

$$\gamma = \frac{g_e}{\bar{g}_e}$$

It is also related to the prediction error as:

$$R(n) = E(n) - \bar{E}(n) = 20 \log \gamma.$$

The codebook search for 4.35, 5.8, 6.65 and 8.0 kbps encoding bit rates consists of two steps. In the first step, a binary search of a single entry table representing the quantized prediction error is performed. In the second step, the index  $Index\_1$  of the optimum entry that is closest to the unquantized prediction error in mean square error sense is used to limit the search of the two-dimensional VQ table representing the adaptive codebook gain and the prediction error. Taking advantage of the particular arrangement and ordering of the VQ table, a fast search using few candidates around the entry pointed by  $Index\_1$  is performed. In fact, only about half of the VQ table entries are tested to lead to the optimum entry with  $Index\_2$ . Only  $Index\_2$  is transmitted.

For 11.0 kbps bit rate encoding mode, a full search of both scalar gain codebooks are used to quantize  $g_e$  and  $g_e$ . For  $g_e$ , the search is performed by minimizing the error

$$Err = abs(g_e - \bar{g}_e).$$

Whereas for  $g_e$ , the search is performed by minimizing the error

$$Err = \left[ \bar{T}_e - \bar{g}_e \bar{C}_e - g_e \bar{C}_e \right]^2.$$

An update of the states of the synthesis and weighting filters is needed in order to compute the target signal for the next subframe. After the two gains are quantized, the excitation signal,  $u(n)$ , in the present subframe is computed as:

$$u(n) = \bar{g}_e v(n) + \bar{g}_e c(n), n = 0, 39,$$

where  $\bar{g}_e$  and  $\bar{g}_e$  are the quantized adaptive and fixed codebook gains respectively,  $v(n)$  the adaptive codebook excitation (interpolated past excitation), and  $c(n)$  is the fixed codebook excitation. The state of the filters can be updated by filtering the signal  $r(n) - u(n)$  through the filters  $1/\bar{\lambda}(z)$  and  $W(z)$  for the 40-sample subframe and saving the states of the filters. This would normally require 3 filterings.

A simpler approach which requires only one filtering is as follows. The local synthesized speech at the encoder,  $\hat{s}(n)$ , is computed by filtering the excitation signal through  $1/\bar{\lambda}(z)$ . The output of the filter due to the input  $r(n) - u(n)$  is equivalent to  $e(n) = \hat{s}(n) - \hat{s}(n)$ , so the states of the synthesis filter  $1/\bar{\lambda}(z)$  are given by  $e(n)$ ,  $n = 0, 39$ . Updating the states of the filter  $W(z)$  can be done by filtering the error signal  $e(n)$  through this filter to find the perceptually weighted error  $e_w(n)$ . However, the signal  $e_w(n)$  can be equivalently found by:

$$e_w(n) = \bar{T}_e(n) - \bar{g}_e C_e(n) - \bar{g}_e C_e(n).$$

The states of the weighting filter are updated by computing  $e_w(n)$  for  $n = 30$  to 39.

The function of the decoder consists of decoding the transmitted parameters (LTP parameters, adaptive codebook vector and its gain, fixed codebook vector and its gain) and performing synthesis to obtain the reconstructed speech. The reconstructed speech is then postfiltered and upsampled.

The decoding process is performed in the following order. First, the LP filter parameters are encoded. The received indices of LSF quantization are used to reconstruct the quantized LSF vector. Interpolation is performed to obtain 4 interpolated LSF vectors (corresponding to 4 subframes). For each subframe, the interpolated LSF vector is converted to LP filter coefficient domain,  $a_i$ , which is used for synthesizing the reconstructed speech in the subframe.

For rates 4.55, 5.8 and 6.65 (during PP\_mode) kbps bit rate encoding modes, the received pitch index is used to interpolate the pitch lag across the entire subframe. The following three steps are repeated for each subframe:

- 1) Decoding of the gains: for bit rates of 4.55, 5.8, 6.65 and 8.0 kbps, the received index is used to find the quantized adaptive codebook gain,  $\bar{g}_a$ , from the 2-dimensional VQ table. The

same index is used to get the fixed codebook gain correction factor  $\gamma$  from the same quantization table. The quantized fixed codebook gain,  $\bar{g}_f$ , is obtained following these

steps:

- the predicted energy is computed  $\bar{E}(n) = \sum_{i=1}^L b_i \hat{R}(n-i)$ ;
- the energy of the unscaled fixed codebook excitation is calculated as  $E_f = 10 \log \left( \frac{1}{40} \sum_{i=1}^{40} c^2(i) \right)$ ; and

-67-

- the predicted gain  $\bar{g}_p$  is obtained as  $\bar{g}_p = 10^{(0.05(\bar{E}(n) - \bar{E}_f))}$ .

The quantized fixed codebook gain is given as  $\bar{g}_f = \gamma \bar{g}_p$ . For 11 kbps bit rate, the received adaptive codebook gain index is used to readily find the quantized adaptive gain,  $\bar{g}_a$ , from the quantization table. The received fixed codebook gain index gives the fixed codebook gain correction factor  $\gamma$ . The calculation of the quantized fixed codebook gain,  $\bar{g}_f$ , follows the same steps as the other rates.

- 2) Decoding of adaptive codebook vector for 8.0, 11.0 and 6.65 (during LTP\_mode=1) kbps bit rate encoding modes, the received pitch index (adaptive codebook index) is used to find the integer and fractional parts of the pitch lag. The adaptive codebook  $w(n)$  is found by interpolating the past excitation  $u(n)$  (at the pitch delay) using the FIR filters.

- 3) Decoding of fixed codebook vector: the received codebook indices are used to extract the type of the codebook (pulse or Gaussian) and either the amplitudes and positions of the excitation pulses or the bases and signs of the Gaussian excitation. In either case, the reconstructed fixed codebook excitation is given as  $c(n)$ . If the integer part of the pitch lag is less than the subframe size 40 and the chosen excitation is pulse type, the pitch sharpening is applied. This translates into modifying  $c(n)$  as  $c(n) = c(n) + \beta c(n-T)$ , where  $\beta$  is the decoded pitch gain  $\bar{g}_p$ , from the previous subframe bounded by [0.2, 1.0].

The excitation at the input of the synthesis filter is given by

$u(n) = \bar{g}_p v(n) + \bar{g}_f c(n), n = 0, 39$ . Before the speech synthesis, a post-processing of the excitation elements is performed. This means that the total excitation is modified by emphasizing the contribution of the adaptive codebook vector.

-68-

$$\bar{u}(n) = \begin{cases} u(n) + 0.25\bar{g}_r u(n), & \bar{g}_r > 0.5 \\ u(n), & \bar{g}_r \leq 0.5 \end{cases}$$

Adaptive gain control (AGC) is used to compensate for the gain difference between the

unemphasized excitation  $u(n)$  and emphasized excitation  $\bar{u}(n)$ . The gain scaling factor  $\eta$  for

the emphasized excitation is computed by:

$$\eta = \begin{cases} \frac{\sum_{n=0}^N \bar{u}^2(n)}{\sum_{n=0}^N u^2(n)}, & \bar{g}_r > 0.5 \\ 1.0, & \bar{g}_r \leq 0.5 \end{cases}$$

The gain-scaled emphasized excitation  $\bar{u}(n)$  is given by:

$$\bar{u}'(n) = \eta \bar{u}(n).$$

The reconstructed speech is given by:

$$\bar{x}(n) = \bar{u}'(n) - \sum_{i=1}^{10} \bar{a}_i \bar{x}(n-i), n = 0 \text{ to } 39,$$

where  $\bar{a}_i$  are the interpolated LP filter coefficients. The synthesized speech  $\bar{x}(n)$  is then passed through an adaptive postfilter.

Post-processing consists of two functions: adaptive postfiltering and signal up-scaling.

The adaptive postfilter is the cascade of three filters: a formant postfilter and two tilt

compensation filters. The postfilter is updated every subframe of 5 ms. The formant postfilter is

given by:

$$H_f(z) = \frac{\bar{A}(z/\gamma_r)}{\bar{A}(z/\gamma_d)}$$

where  $\bar{A}(z)$  is the received quantized and interpolated LP inverse filter and  $\gamma_r$  and  $\gamma_d$  control the amount of the formant postfiltering.

The first tilt compensation filter  $H_{t1}(z)$  compensates for the tilt in the formant postfilter

$H_f(z)$  and is given by:

$$H_{t1}(z) = (1 - \mu z^{-1})$$

where  $\mu = \gamma_r k_1$  is a tilt factor, with  $k_1$  being the first reflection coefficient calculated on the truncated impulse response  $h_f(n)$ , of the formant postfilter  $k_1 = \frac{r_1(1)}{r_1(0)}$  with:

$$r_1(i) = \sum_{j=0}^{L-1-i} h_f(j)h_f(j+i), (L_r = 22).$$

The postfiltering process is performed as follows. First, the synthesized speech  $\bar{x}(n)$  is

inverse filtered through  $\bar{A}(z/\gamma_r)$  to produce the residual signal  $\bar{r}(n)$ . The signal  $\bar{r}(n)$  is filtered

by the synthesis filter  $1/\bar{A}(z/\gamma_d)$  is passed to the first tilt compensation filter  $h_{t1}(z)$  resulting in the postfiltered speech signal  $\bar{x}_f(n)$ .

Adaptive gain control (AGC) is used to compensate for the gain difference between the

synthesized speech signal  $\bar{x}(n)$  and the postfiltered signal  $\bar{x}_f(n)$ . The gain scaling factor  $\gamma$  for the present subframe is computed by:

$$\gamma = \sqrt{\frac{\sum_{n=0}^N \bar{x}^2(n)}{\sum_{n=0}^N \bar{x}_f^2(n)}}$$

The gain-scaled postfiltered signal  $\bar{x}'(n)$  is given by:

$$\bar{x}'(n) = \beta(n)\bar{x}_f(n)$$

where  $\beta(n)$  is updated in sample by sample basis and given by:

$$\beta(n) = \alpha\beta(n-1) + (1-\alpha)\gamma$$

where  $\alpha$  is an AGC factor with value 0.9. Finally, up-scaling consists of multiplying the postfiltered speech by a factor 2 to undo the down scaling by 2 which is applied to the input signal.

Figs. 6 and 7 are drawings of an alternate embodiment of a 4 kbps speech codec that also illustrates various aspects of the present invention. In particular, Fig. 6 is a block diagram of a speech encoder 601 that is built in accordance with the present invention. The speech encoder 601 is based on the analysis-by-synthesis principle. To achieve toll quality at 4 kbps, the speech encoder 601 departs from the strict waveform-matching criterion of regular CELP coders and strives to catch the perceptual important features of the input signal.

The speech encoder 601 operates on a frame size of 20 ms with three subframes (two of 6.625 ms and one of 6.75 ms). A look-ahead of 15 ms is used. The one-way coding delay of the codec adds up to 55 ms.

At a block 615, the spectral envelope is represented by a  $10^{\text{th}}$  order LPC analysis for each frame. The prediction coefficients are transformed to the Line Spectrum Frequencies (LSFs) for quantization. The input signal is modified to better fit the coding model without loss of quality. This processing is denoted "signal modification" as indicated by a block 621. In order to improve the quality of the reconstructed signal, perceptual important features are estimated and emphasized during encoding.

The excitation signal for an LPC synthesis filter 625 is build from the two traditional components: 1) the pitch contribution; and 2) the innovation contribution. The pitch contribution is provided through use of an adaptive codebook 627. An innovation codebook 629 has several

-71-

subcodebooks in order to provide robustness against a wide range of input signals. To each of the two contributions a gain is applied which, multiplied with their respective codebook vectors and summed, provide the excitation signal.

The LSFs and pitch lag are coded on a frame basis, and the remaining parameters (the innovation codebook index, the pitch gain, and the innovation codebook gain) are coded for every subframe. The LSF vector is coded using predictive vector quantization. The pitch lag has an integer part and a fractional part constituting the pitch period. The quantized pitch period has a non-uniform resolution with higher density of quantized values at lower delays. The bit allocation for the parameters is shown in the following table.

Table of Bit Allocation

Parameter	Bits per 20 ms
LSFs	21
Pitch lag (adaptive codebook)	8
Gains	12
Innovation codebook	3x13 = 39
Total	80

When the quantization of all parameters for a frame is complete the indices are multiplexed to form the 80 bits for the serial bit-stream.

Fig. 7 is a block diagram of a decoder 701 with corresponding functionality to that of the encoder of Fig. 6. The decoder 701 receives the 80 bits on a frame basis from a demultiplexor 711. Upon receipt of the bits, the decoder 701 checks the sync-word for a bad frame indication, and decides whether the entire 80 bits should be disregarded and frame erasure concealment applied. If the frame is not declared a frame erasure, the 80 bits are mapped to the parameter indices of the codec, and the parameters are decoded from the indices using the inverse quantization schemes of the encoder of Fig. 6.

-72-

When the LSFs, pitch lag, pitch gains, innovation vectors, and gains for the innovation vectors are decoded, the excitation signal is reconstructed via a block 715. The output signal is synthesized by passing the reconstructed excitation signal through an LPC synthesis filter 721. To enhance the perceptual quality of the reconstructed signal both short-term and long-term post-processing are applied at a block 731.

Regarding the bit allocation of the 4 kbps codec (as shown in the prior table), the LSFs and pitch lag are quantized with 21 and 8 bits per 20 ms, respectively. Although the three subframes are of different size the remaining bits are allocated evenly among them. Thus, the innovation vector is quantized with 13 bits per subframe. This adds up to a total of 80 bits per 20 ms, equivalent to 4 kbps.

The estimated complexity numbers for the proposed 4 kbps codec are listed in the following table. All numbers are under the assumption that the codec is implemented on commercially available 16-bit fixed point DSPs in full duplex mode. All storage numbers are under the assumption of 16-bit words, and the complexity estimates are based on the floating point C-source code of the codec.

Table of Complexity Estimates

Computational complexity	30 MIPS
Program and data ROM	18 kwords
RAM	3 kwords

The decoder 701 comprises decode processing circuitry that generally operates pursuant to software control. Similarly, the encoder 601 (Fig. 6) comprises encoder processing circuitry also operating pursuant to software control. Such processing circuitry may coexist, at least in part, within a single processing unit such as a single DSP.

Fig. 8a is a timing diagram of an exemplary pitch lag contour over two speech frames to which continuous warping techniques are applied in accordance with the present invention. In particular, an exemplary pitch lag contour, an original pitch lag contour 811, typically varies rather slowly over time. From a beginning of a first frame, as indicated by a marker 813, the original pitch lag contour 811 varies generally upward through a plurality of subframes, as indicated by subframe markers 819 and 821. Similarly, the upward trend can be seen in a second frame ending at a marker 811.

Without applying warping of the present invention, it can be appreciated that the amount of bits needed to code the original pitch lag contour 811 might prove excessive, especially at the lower encoder bit rates. Moreover, any attempt to search for a match of such pitch contour, such as shifting each of the pitch pulses in an original residual, proves difficult and requires reliable endpoint detection to maintain signal continuity.

Fig. 8b is a timing diagram illustrating a linear pitch contour to which continuous warping of the original pitch lag contour is applied in accordance with the present invention. Specifically, a linear segment 831 for a first frame, a linear segment 833 for a second frame, etc., provide a basis for warping the pitch lag contour 811. By performing continuous warping, the pitch contour 811 is effectively compressed during some periods, e.g., at a time period 835, and expanded during others, e.g., during a time period 837 to match the contour defined by the segments 831, 833, and so on.

From frame to frame such warping takes place, i.e., continuous warping is applied. Such processing or portions thereof might take place on subframe, multiple subframe, multiple frame basis, or other time period, for example. Similarly, although only three subframes are shown, more or less might be used with equal or unequal time period definition.

The warping to conform the pitch lag contour defined by the segments 831 and 833, for example, may be applied to the residual speech signal in an open loop approach. Alternatively, in some embodiments such as the specific embodiment described above in reference to Figs. 2-4, continuous warping is applied to the weighted speech signal (although the original speech signal might alternatively have been used) in a closed loop fashion. Searching for the best match can be performed rapidly by finding the optimal end of the original (weighted or residual) signal with a limited range to make the modified signal match the new pitch contour.

Fig. 8c is a diagram illustrating the use of the new pitch contour of Fig. 8b which can be represented by a lesser number of bits than the original pitch contour of Fig. 8a. A new pitch contour 841 comprising the linear segments 831 and 833 is defined by encoding the pitch lag at each segment marker. Having received such coding information, the decoder can reconstruct intermediate pitch lag values merely through interpolation, for example, as indicated at the subframe markers.

Fig. 9 is a flow diagram illustrating an embodiment of the continuous warping approach and an associated fast searching process used by an encoder of the present invention to carry out the functionality described in reference to Figs. 8a-c on a residual signal using an open loop approach. At a block 909, the encoder, i.e., the encoder processing circuitry operating pursuant to software instruction, first identifies maps the original residual to the modified residual, i.e., the original residual is mapped to a linear pitch contour defined by a previous and a current frame pitch lag value.

Specifically, at the block 909, the original residual having a  $T_{\text{start}}$  and a  $T_{\text{end}}$  is mapped to a modified residual defined by a  $T_{\text{start}}$  and a  $T_{\text{end}}$ . Thereafter, at a block 913, the encoder identifies a range in which an optimal value of  $T_{\text{end}}$  is searched. The search is performed at a

block 917 to make the modified residual best fit the pitch contour. With the optimal endpoint  $T_{\text{end}}$  found, at a block 921, the original residual is warped from the  $T_{\text{start}}$  and the optimal  $T_{\text{end}}$  to the modified residual ( $T_{\text{start}}$  and  $T_{\text{end}}$ ) as follows:

$$\begin{aligned} T_{\text{start}} &= T_{\text{start}} + L_n \\ T_{\text{start}} &= T_{\text{start}} + L \cdot (T_{\text{end}} - T_{\text{start}}) / (T_{\text{end}} - T_{\text{start}}), \end{aligned}$$

where  $L$  comprises the working step size.

Fig. 10 is a flow diagram illustrating an alternate embodiment of functionality of a speech encoder of the present invention that performs continuous warping to the weighted speech signal in a closed loop approach. In particular, at a block 1011, the encoder estimates pitch lag at the end of a frame. Such estimation is based on the normalized correlation:

$$R_k = \frac{\sum_{n=0}^L s_n(n+nl) s_n(n+nl-k)}{\sqrt{\sum_{n=0}^L s_n^2(n+nl-k)}}$$

where  $s_n(n+nl)$ ,  $n = 0, 1, \dots, L-1$ , represents the last segment of the weighted speech signal including the look-ahead (the look-ahead length is 25 samples), and the size  $L$  is defined according to the open-loop pitch lag  $T_{op}$  with the corresponding normalized correlation  $C_{T_{op}}$ :

$$\begin{aligned} \text{if } (C_{T_{op}} > 0.6) \\ L &= \max(50, T_{op}) \\ L &= \min(80, L) \\ \text{else} \\ L &= 80 \end{aligned}$$

To identify the pitch lag estimate, the encoder first selects one integer lag  $k$  maximizing the  $R_k$  in the range  $k \in [T_{op} - 10, T_{op} + 10]$  bounded by [17, 145]. Then, the precise pitch lag  $P_m$  and the corresponding index  $L_m$  for the current frame is searched around the integer lag,  $|k-l|$ .

$k+1$ ), by up-sampling  $R_k$ . The possible candidates for the pitch lag are obtained from the table named as *PitLagTab8b*( $l$ ),  $l=0,1,\dots,127$ . Lastly, the pitch lag  $P_m = \text{PitLagTab8b}(l_m)$  is possibly modified by checking the accumulated delay  $\tau_{acc}$  due to the modification of the speech signal:

if ( $\tau_{acc} > 5$ )  $l_m \leftarrow \min\{l_m + 1, 127\}$ ,  
 if ( $\tau_{acc} < -5$ )  $l_m \leftarrow \max\{l_m - 1, 0\}$ ;

it could be modified again:

if ( $\tau_{acc} > 10$ )  $l_m \leftarrow \min\{l_m + 1, 127\}$ ,  
 if ( $\tau_{acc} < -10$ )  $l_m \leftarrow \max\{l_m - 1, 0\}$ ;

The obtained index  $l_m$  will be sent to the decoder.

At a block 1013, the pitch lag contour,  $\tau_c(n)$ , is identified using both the current pitch

lag  $P_m$  and the previous pitch lag  $P_{m-1}$ :

if ( $|P_m - P_{m-1}| < 0.2 \min(P_m, P_{m-1})$ )  
 $\tau_c(n) = P_{m-1} + n(P_m - P_{m-1})/L_f$ ,  $n = 0,1,\dots,L_f - 1$   
 $\tau_c(n) = P_m$ ,  $n = L_f, \dots, 170$   
 else

$\tau_c(n) = P_{m-1}$ ,  $n = 0,1,\dots,39$ ;

$\tau_c(n) = P_m$ ,  $n = 40, \dots, 170$

where  $L_f = 160$  is the frame size.

In the present embodiment, each frame is divided into 3 subframes for the long-term

preprocessing. For the first two subframes, the subframe size,  $L_s$ , is 53, and the subframe size for searching,  $L_{sr}$ , is 70. For the last subframe,  $L_f$  is 54 and  $L_{sr}$  is:

$$L_{sr} = \min(70, L_f + L_{sld} - 10 - \tau_{acc}),$$

where  $L_{sld}=25$  is the look-ahead and the maximum of the accumulated delay  $\tau_{acc}$  is limited to

14.

At a block 1015, the weighted speech signal is mapped to the pitch lag contour,  $\tau_c(n)$ .

In particular, the target for the modification process of the weighted speech, temporally

memorized in  $\{\hat{s}_w(m0+n), n = 0,1,\dots,L_{sr}-1\}$  is calculated by mapping, i.e., warping, the past modified weighted speech buffer,  $\hat{s}_w(m0+n)$ ,  $n < 0$ , with the pitch lag contour,

$$\tau_c(n+m \cdot L_f), m = 0,1,2,$$

$$\hat{s}_w(m0+n) = \sum_{i=m0}^{\tau_c(n)} \hat{s}_w(m0+n - T_c(n) + i) I_f(i, T_c(n)), n = 0,1,\dots,L_{sr}-1,$$

where  $T_c(n)$  and  $T_{fc}(n)$  are calculated by

$$T_c(n) = \text{trunc}(\tau_c(n+m \cdot L_f)),$$

$$T_{fc}(n) = \tau_c(n) - T_c(n),$$

$m$  is subframe number,  $I_f(i, T_{fc}(n))$  is a set of interpolation coefficients, and  $f_i$  is 10. Then, the

target for matching,  $\hat{s}_f(n)$ ,  $n = 0,1,\dots,L_{sr}-1$ , is calculated by weighting

$\hat{s}_w(m0+n)$ ,  $n = 0,1,\dots,L_{sr}-1$ , in the time domain:

$$\hat{s}_f(n) = n \cdot \hat{s}_w(m0+n) / L_{sr}, n = 0,1,\dots,L_{sr}-1,$$

$$\hat{s}_f(n) = \hat{s}_w(m0+n), n = L_{sr}, \dots, L_{sr}-1,$$

At a block 1017, the encoder calculates a relatively small shift range for seeking the best local delay. Specifically, the local integer shifting range ( $SR0, SRI$ ) for searching for the best

local delay is computed as the following:

if speech is unvoiced

$$SR0 = -1,$$

$$SRI = 1,$$

else

$$SR0 = \text{round}[-4 \min(1.0, \max(0.0, 1-0.4(P_{sh}-0.2)))],$$

$$SRI = \text{round}[4 \min(1.0, \max(0.0, 1-0.4(P_{sh}-0.2)))]],$$

where  $P_{sh} = \max(P_{shl}, P_{shh})$ ,  $P_{shl}$  is the average to peak ratio (i.e., sharpness) from the target

signal:

$$P_{sh} = \frac{\sum_{n=0}^{L_{sr}-1} |\hat{s}_w(m0+n)|}{L_{sr} \max(|\hat{s}_w(m0+n)|, n = 0,1,\dots,L_{sr}-1)}$$

and  $P_{m2}$  is the sharpness from the weighted speech signal,

$$P_{m2} = \frac{(L_p - L_f / 2) \max_n \left| \sum_{n=0}^{L_p - L_f / 2 - 1} \{x_p(n + n0 + L_f / 2)\} \right|}{\sum_{n=0}^{L_p - L_f / 2 - 1} \{x_p(n + n0 + L_f / 2)\}}$$

where  $n0 = \text{trunc}(m0 + \tau_{acc} + 0.5)$  (here,  $m$  is subframe number and  $\tau_{acc}$  is the previous accumulated delay).

At a block 1019, the encoder searches for then adjusts the best local delay. Such searching involves use of linear time weighting. In particular, to find the best local delay,  $\tau_{opt}$ , at the end of the current processing subframe, a normalized correlation vector between the weighted speech signal and the modified matching target is defined as:

$$R_l(k) = \frac{\sum_{n=0}^{L_p-1} x_w(n0 + n + k) \hat{z}_l(n)}{\sqrt{\sum_{n=0}^{L_p-1} x_w^2(n0 + n + k) \sum_{n=0}^{L_p-1} \hat{z}_l^2(n)}}$$

A best local delay in the integer domain,  $k_{opt}$ , is selected by maximizing  $R_l(k)$  in the range of  $k \in [SFO, SRI]$ , which is corresponding to the real delay:

$$k_r = k_{opt} + n0 - m0 - \tau_{acc}$$

If  $R_l(k_{opt}) < 0.5$ ,  $k_r$  is set to zero.

In order to get a more precise local delay in the range  $[k_r - 0.75 + 0.1j, j=0, 1, \dots, 15]$  around  $k_r$ ,  $R_l(k)$  is interpolated to obtain the fractional correlation vector,  $R_f(j)$ , which is given by:

$$R_f(j) = \sum_{i=0}^1 R_l(k_{opt} + j_i + i) I_f(i, j), \quad j=0, 1, \dots, 15,$$

where  $\{I_f(i, j)\}$  is a set of interpolation coefficients. The optimal fractional delay index,  $j_{opt}$ , is selected by maximizing  $R_f(j)$ . Finally, the best local delay,  $\tau_{opt}$ , at the end of the current processing subframe, is given:

$$\tau_{opt} = k_r - 0.75 + 0.1 j_{opt}$$

Once found, the best local delay is then adjusted as follows.

$$\tau_{opt} = \begin{cases} 0, & \text{if } \tau_{acc} + \tau_{opt} > 14 \\ \tau_{opt}, & \text{otherwise} \end{cases}$$

At a block 1021, the original weighted speech is warped from an original to a modified time region. Specifically, the modified weighted speech of the current subframe, memorized in  $\{\hat{z}_p(m0 + n), n=0, 1, \dots, L_p - 1\}$  to update the buffer and produce the target for the fixed codebook search, is generated by warping the original weighted speech  $\{x_p(n)\}$  from the original time region:

$$\{m0 + \tau_{acc}, m0 + \tau_{acc} + L_p + \tau_{opt}\},$$

to the modified time region,

$$\{m0, m0 + L_p\}:$$

$$\hat{z}_p(m0 + n) = \sum_{i=m0}^{L_p} x_p(m0 + n + T_w(n) + i) I_p(i, T_w(n)), \quad n=0, 1, \dots, L_p - 1,$$

where  $T_w(n)$  and  $T_w(n)$  are calculated by:

$$\begin{aligned} T_w(n) &= \text{trunc}(\tau_{acc} + n \cdot \tau_{opt} / L_p), \\ T_w(n) &= \tau_{acc} + n \cdot \tau_{opt} / L_p - T_w(n), \end{aligned}$$

$\{I_p(i, T_w(n))\}$  is a set of interpolation coefficients.

To complete the process after having completed the warping of the weighted speech for the current subframe, the modified target weighted speech buffer is updated as follows:

$$\hat{z}_p(n) \Leftarrow \hat{z}_p(n + L_p), \quad n=0, 1, \dots, n_m - 1$$



The accumulated delay at the end of the current subframe is renewed by:

$$\tau_{acc} \Leftarrow \tau_{acc} + \tau_{opt}$$

As previously articulated, although the continuous warping processes described with reference to Fig. 10 is applied to the weighted speech signal, it might alternatively be applied to the residual or, for example, to the original unweighted speech signal.

Of course, many other modifications and variations are also possible. In view of the above detailed description of the present invention and associated drawings, such other modifications and variations will now become apparent to those skilled in the art. It should also be apparent that such other modifications and variations may be effected without departing from the spirit and scope of the present invention.

In addition, the following Appendix A provides a list of many of the definitions, symbols and abbreviations used in this application. Appendices B and C respectively provide source and channel bit ordering information at various encoding bit rates used in one embodiment of the present invention. Appendices A, B and C comprise part of the detailed description of the present application, and, otherwise, are hereby incorporated herein by reference in its entirety.

## APPENDIX A

For purposes of this application, the following symbols, definitions and abbreviations apply.

adaptive codebook:

The adaptive codebook contains excitation vectors that are adapted for every subframe. The adaptive codebook is derived from the long term filter state. The pitch lag value can be viewed as an index into the adaptive codebook.

adaptive postfilter:

The adaptive postfilter is applied to the output of the short term synthesis filter to enhance the perceptual quality of the reconstructed speech. In the adaptive multi-rate codec (AMR), the adaptive postfilter is a cascade of two filters: a formant postfilter and a tilt compensation filter.

Adaptive Multi Rate codec:

The adaptive multi-rate code (AMR) is a speech and channel codec capable of operating at gross bit-rates of 11.4 kbps ("half-rate") and 22.8 kbps ("full-rate"). In addition, the codec may operate at various combinations of speech and channel coding (codec mode) bit-rates for each channel mode.

AMR handover:

Handover between the full rate and half rate channel modes to optimize AMR operation.

channel mode:

Half-rate (HR) or full-rate (FR) operation.

channel mode adaptation:

The control and selection of the (FR or HR) channel mode.

channel repacking:

Repacking of HR (and FR) radio channels of a given radio cell to achieve higher capacity within the cell.

closed-loop pitch analysis:

This is the adaptive codebook search, i.e., a process of estimating the pitch (lag) value from the weighted input speech and the long term filter state. In the closed-loop search, the lag is searched using error minimization loop (analysis-by-synthesis). In the adaptive multi rate codec, closed-loop pitch search is performed for every subframe.

codec mode:

For a given channel mode, the bit partitioning between the speech and channel codecs.

codec mode adaptation:

The control and selection of the codec mode bit-rates. Normally, implies no change to the channel mode.

direct form coefficients:	One of the formats for storing the short term filter parameters. In the adaptive multi rate codec, all filters used to modify speech samples use direct form coefficients.
fixed codebook:	The fixed codebook contains excitation vectors for speech synthesis filters. The contents of the codebook are non-adaptive (i.e., fixed). In the adaptive multi rate codec, the fixed codebook for a specific rate is implemented using a multi-function codebook.
fractional lags:	A set of lag values having sub-sample resolution. In the adaptive multi rate codec a sub-sample resolution between $1/6^{\text{th}}$ and 1.0 of a sample is used.
full-rate (FR):	Full-rate channel or channel mode.
frame:	A time interval equal to 20 ms (160 samples at an 8 kHz sampling rate).
gross bit-rate:	The bit-rate of the channel mode selected (22.8 kbps or 11.4 kbps).
half-rate (HR):	Half-rate channel or channel mode.
in-band signaling:	Signaling for DTX, Link Control, Channel and codec mode modification, etc. carried within the traffic.
integer lags:	A set of lag values having whole sample resolution.
interpolating filter:	An FIR filter used to produce an estimate of sub-sample resolution samples, given an input sampled with integer sample resolution.
inverse filter:	This filter removes the short term correlation from the speech signal. The filter models an inverse frequency response of the vocal tract.
lag:	The long term filter delay. This is typically the true pitch period, or its multiple or sub-multiple.
Line Spectral Frequencies:	(see Line Spectral Pair)
Line Spectral Pair:	Transformation of LPC parameters. Line Spectral Pairs are obtained by decomposing the inverse filter transfer function $A(z)$ to a set of two transfer functions, one having even symmetry and the other having odd symmetry. The Line Spectral Pairs (also called as Line Spectral Frequencies) are the roots of these polynomials on the z-unit circle).

LP analysis window:	For each frame, the short term filter coefficients are computed using the high pass filtered speech samples within the analysis window. In the adaptive multi rate codec, the length of the analysis window is always 240 samples. For each frame, two asymmetric windows are used to generate two sets of LP coefficient coefficients which are interpolated in the LSF domain to construct the perceptual weighting filter. Only a single set of LP coefficients per frame is quantized and transmitted to the decoder to obtain the synthesis filter. A lookahead of 25 samples is used for both HR and FR.
LP coefficients:	Linear Prediction (LP) coefficients (also referred as Linear Predictive Coding (LPC) coefficients) is a generic descriptive term for describing the short term filter coefficients.
LTP Mode:	Codec works with traditional LTP.
mode:	When used alone, refers to the source codec mode, i.e., to one of the source codecs employed in the AMR codec. (See also codec mode and channel mode.)
multi-function codebook:	A fixed codebook consisting of several subcodebooks constructed with different kinds of pulse innovation vector structures and noise innovation vectors, where codeword from the codebook is used to synthesize the excitation vectors.
open-loop pitch search:	A process of estimating the near optimal pitch lag directly from the weighted input speech. This is done to simplify the pitch analysis and confine the closed-loop pitch search to a small number of lags around the open-loop estimated lags. In the adaptive multi rate codec, open-loop pitch search is performed once per frame for PP mode and twice per frame for LTP mode.
out-of-band signaling:	Signaling on the GSM control channels to support link control.
PP Mode:	Codec works with pitch preprocessing.
residual:	The output signal resulting from an inverse filtering operation.
short term synthesis filter:	This filter introduces, into the excitation signal, short term correlation which models the impulse response of the vocal tract.
perceptual weighting filter:	This filter is employed in the analysis-by-synthesis search of the codebooks. The filter exploits the noise masking properties of the formants (vocal tract resonances) by weighting the error less in regions near the formant frequencies and more in regions away from them.

subframe:	A time interval equal to 5-10 ms (40-80 samples at an 8 kHz sampling rate).		
vector quantization:	A method of grouping several parameters into a vector and quantizing them simultaneously.	$H_f(z)$	Tilt compensation filter
zero input response:	The output of a filter due to past inputs, i.e. due to the present state of the filter, given that an input of zeros is applied.	$\gamma_i$	Control coefficient for the amount of the tilt compensation filtering
zero state response:	The output of a filter due to the present input, given that no past inputs have been applied, i.e., given the state information in the filter is all zeroes.	$\mu = \gamma_i k_i$	A tilt factor, with $k_i$ being the first reflection coefficient
$A(z)$	The inverse filter with unquantized coefficients	$h_f(n)$	The truncated impulse response of the formant postfilter
$\hat{A}(z)$	The inverse filter with quantized coefficients	$L_n$	The length of $h_f(n)$
$H(z) = \frac{1}{\hat{A}(z)}$	The speech synthesis filter with quantized coefficients	$r_k(i)$	The auto-correlations of $h_f(n)$
$a_i$	The unquantized linear prediction parameters (direct form coefficients)	$\hat{A}(z/\gamma_n)$	The inverse filter (numerator) part of the formant postfilter
$\hat{a}_i$	The quantized linear prediction parameters	$1/\hat{A}(z/\gamma_d)$	The synthesis filter (denominator) part of the formant postfilter
$\frac{1}{B(z)}$	The long-term synthesis filter	$\hat{r}(n)$	The residual signal of the inverse filter $\hat{A}(z/\gamma_n)$
$W(z)$	The perceptual weighting filter (unquantized coefficients)	$h_f(z)$	Impulse response of the tilt compensation filter
$\gamma_1, \gamma_2$	The perceptual weighting factors	$\beta_{ac}(n)$	The AGC-controlled gain scaling factor of the adaptive postfilter
$F_2(z)$	Adaptive pre-filter	$\alpha$	The AGC factor of the adaptive postfilter
$T$	The nearest integer pitch lag to the closed-loop fractional pitch lag of the subframe	$H_{ad}(z)$	Pre-processing high-pass filter
$\beta$	The adaptive pre-filter coefficient (the quantized pitch gain)	$w_f(n), w_{II}(n)$	LP analysis windows
$H_f(z) = \frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)}$	The formant postfilter	$L_1^{(I)}$	Length of the first part of the LP analysis window $w_f(n)$
$\gamma_n$	Control coefficient for the amount of the formant post-filtering	$L_2^{(I)}$	Length of the second part of the LP analysis window $w_f(n)$
$\gamma_d$	Control coefficient for the amount of the formant post-filtering	$L_1^{(II)}$	Length of the first part of the LP analysis window $w_{II}(n)$
		$L_2^{(II)}$	Length of the second part of the LP analysis window $w_{II}(n)$
		$r_{ac}(k)$	The auto-correlations of the windowed speech $s(n)$
		$w_{ac}(i)$	Lag window for the auto-correlations (60 Hz bandwidth expansion)
		$f_0$	The bandwidth expansion in Hz

$f_s$	The sampling frequency in Hz	$f' = [f_1 f_2 \dots f_{10}]$	The vector representation of the LSFs in Hz
$r'_{ac}(k)$	The modified (bandwidth expanded) auto-correlations	$x^{(1)}/(n), x^{(2)}/(n)$	The mean-removed LSF vectors at frame $n$
$E_{LP}(i)$	The prediction error in the $i$ th iteration of the Levinson algorithm	$r^{(1)}/(n), r^{(2)}/(n)$	The LSF prediction residual vectors at frame $n$
$k_i$	The $i$ th reflection coefficient	$p(n)$	The predicted LSF vector at frame $n$
$a_j^{(i)}$	The $j$ th direct form coefficient in the $i$ th iteration of the Levinson algorithm	$r^{(2)}/(n-1)$	The quantized second residual vector at the past frame
$F_1(z)$	Symmetric LSF polynomial	$\hat{p}^k$	The quantized LSF vector at quantization index $k$
$F_2(z)$	Antisymmetric LSF polynomial	$E_{LSP}$	The LSF quantization error
$\hat{F}_1(z)$	Polynomial $\hat{F}_1(z)$ with root $z = -1$ eliminated	$w_i, i = 1, \dots, 10$	LSF-quantization weighting factors
$\hat{F}_2(z)$	Polynomial $\hat{F}_2(z)$ with root $z = 1$ eliminated	$d_i$	The distance between the line spectral frequencies $f_{i+1}$ and $f_{i-1}$
$q_i$	The line spectral pairs (LSFs) in the cosine domain	$M(n)$	The impulse response of the weighted synthesis filter
$q$	An LSF vector in the cosine domain	$O_k$	The correlation maximum of open-loop pitch analysis at delay $k$
$\hat{q}^{(n)}$	The quantized LSF vector at the $n$ th subframe of the frame $n$	$O_i, i = 1, \dots, 3$	The correlation maxima at delays $i, i = 1, \dots, 3$
$\omega_i$	The line spectral frequencies (LSFs)	$(M_i, i_i), i = 1, \dots, 3$	The normalized correlation maxima $M_i$ and the corresponding delays $i_i, i = 1, \dots, 3$
$T_n(x)$	A $n$ th order Chebyshev polynomial	$H(z)W(z) = \frac{A(z/Y_1)}{A(z)A(z/Y_2)}$	The weighted synthesis filter
$f_1(0), f_2(0)$	The coefficients of the polynomials $F_1(z)$ and $F_2(z)$	$A(z/Y_1)$	The numerator of the perceptual weighting filter
$\hat{f}_1(0), \hat{f}_2(0)$	The coefficients of the polynomials $\hat{F}_1(z)$ and $\hat{F}_2(z)$	$1/A(z/Y_2)$	The denominator of the perceptual weighting filter
$f(i)$	The coefficients of either $F_1(z)$ or $F_2(z)$	$T_1$	The nearest integer to the fractional pitch lag of the previous (1st or 3rd) subframe
$C(x)$	Sum polynomial of the Chebyshev polynomials	$s'(n)$	The windowed speech signal
$x$	Cosine of angular frequency $\omega$	$s_c(n)$	The weighted speech signal
$\lambda_k$	Recursion coefficients for the Chebyshev polynomial evaluation	$\hat{s}(n)$	Reconstructed speech signal
$f_i$	The line spectral frequencies (LSFs) in Hz		

$\hat{x}'(n)$	The gain-scaled post-filtered signal	$\mathbf{d} = \mathbf{H}'\mathbf{x}_2$	The correlation between the target signal $\mathbf{x}_2(n)$ and the impulse response $h(n)$ , i.e., backward filtered target
$\hat{x}_f(n)$	Post-filtered speech signal (before scaling)	$\mathbf{H}$	The lower triangular Toeplitz convolution matrix with diagonal $h(0)$ and lower diagonals $h(1), \dots, h(39)$
$\mathbf{x}(n)$	The target signal for adaptive codebook search	$\Phi = \mathbf{H}'\mathbf{H}$	The matrix of correlations of $h(n)$
$\mathbf{x}_2(n), \mathbf{x}_2'$	The target signal for Fixed codebook search	$d(n)$	The elements of the vector $\mathbf{d}$
$res_{LP}(n)$	The LP residual signal	$\phi(i, l)$	The elements of the symmetric matrix $\Phi$
$\mathbf{c}(n)$	The fixed codebook vector	$\mathbf{c}_i$	The innovation vector
$\mathbf{v}(n)$	The adaptive codebook vector	$\mathbf{C}$	The correlation in the numerator of $A_k$
$\mathbf{y}(n) = \mathbf{v}(n) * h(n)$	The filtered adaptive codebook vector	$m_i$	The position of the $i$ th pulse
$\mathbf{y}_f(n)$	The filtered fixed codebook vector	$\phi_i$	The amplitude of the $i$ th pulse
$\mathbf{u}(n)$	The past filtered excitation	$N_p$	The number of pulses in the fixed codebook excitation
$\hat{\mathbf{u}}(n)$	The excitation signal	$E_D$	The energy in the denominator of $A_k$
$\tilde{\mathbf{u}}(n)$	The fully quantized excitation signal	$res_{LTP}(n)$	The normalized long-term prediction residual
$T_{op}$	The gain-scaled emphasized excitation signal	$b(n)$	The sum of the normalized $d(n)$ vector and normalized long-term prediction residual $res_{LTP}(n)$
$t_{min}$	The best open-loop lag	$s_b(n)$	The sign signal for the algebraic codebook search
$t_{max}$	Minimum lag search value	$\mathbf{z}', \mathbf{z}(n)$	The fixed codebook vector convolved with $h(n)$
$R(k)$	Maximum lag search value	$E(n)$	The mean-removed innovation energy (in dB)
$R(k)_i$	Correlation term to be maximized in the adaptive codebook search	$\bar{E}$	The mean of the innovation energy
$A_k$	The interpolated value of $R(k)$ for the integer delay $k$ and fraction $i$	$\bar{E}(n)$	The predicted energy
$C_k$	Correlation term to be maximized in the algebraic codebook search at index $k$	$[b_1 b_2 b_3 b_4]$	The MA prediction coefficients
$E_k$	The correlation in the numerator of $A_k$ at index $k$	$\hat{R}(k)$	The quantized prediction error at subframe $k$
$E_{Dk}$	The energy in the denominator of $A_k$ at index $k$		

$E_i$	The mean innovation energy	HR	Half Rate
$R(n)$	The prediction error of the fixed-codebook gain quantization		
$E_q$	The quantization error of the fixed-codebook gain quantization	LP	Linear Prediction
$e(n)$	The states of the synthesis filter $1/\hat{A}(z)$	LPC	Linear Predictive Coding
$e_w(n)$	The perceptually weighted error of the analysis-by-synthesis search	LSF	Line Spectral Frequency
$\eta$	The gain scaling factor for the emphasized excitation	LSF	Line Spectral Pair
$\delta_c$	The gain scaling factor for the emphasized excitation	LTP	Long Term Predictor (or Long Term Prediction)
$\delta_c$	The fixed-codebook gain	MA	Moving Average
$\hat{\delta}_c$	The predicted fixed-codebook gain	TFO	Tandem Free Operation
$\hat{\delta}_c$	The quantized fixed codebook gain	VAD	Voice Activity Detection
$\delta_a$	The adaptive codebook gain		
$\hat{\delta}_a$	The quantized adaptive codebook gain		
$\gamma_{sc} = \delta_c / \hat{\delta}_c$	A correction factor between the gain $\delta_c$ and the estimated one $\hat{\delta}_c$		
$\hat{\gamma}_{sc}$	The optimum value for $\gamma_{sc}$		
$\gamma_{sc}$	Gain scaling factor		
AGC	Adaptive Gain Control		
AMR	Adaptive Multi Rate		
CELP	Code Excited Linear Prediction		
CI	Carrier-to-Interferer ratio		
DTX	Discontinuous Transmission		
EFR	Enhanced Full Rate		
FIR	Finite Impulse Response		
FR	Full Rate		

# APPENDIX B

## Bit ordering (source coding)

Bit ordering of output bits from source encoder (11 bits/s).

Bits	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19-24	Index of 4 <sup>th</sup> LSF stage
25-32	Index of 5 <sup>th</sup> LSF stage
33-37	Index of adaptive codebook gain, 1 <sup>st</sup> subframe
38-41	Index of adaptive codebook gain, 1 <sup>st</sup> subframe
42-46	Index of adaptive codebook gain, 2 <sup>nd</sup> subframe
47-50	Index of adaptive codebook gain, 2 <sup>nd</sup> subframe
51-55	Index of adaptive codebook gain, 3 <sup>rd</sup> subframe
56-59	Index of adaptive codebook gain, 3 <sup>rd</sup> subframe
60-64	Index of adaptive codebook gain, 4 <sup>th</sup> subframe
65-73	Index of adaptive codebook gain, 4 <sup>th</sup> subframe
74-87	Index of adaptive codebook, 1 <sup>st</sup> subframe
88-94	Index of adaptive codebook, 1 <sup>st</sup> subframe
95-96	Index of adaptive codebook (relative), 2 <sup>nd</sup> subframe
97-100	Index for LSF interpolation
101-108	Index for fixed codebook, 1 <sup>st</sup> subframe
109-119	Index for fixed codebook, 2 <sup>nd</sup> subframe
120-129	Index for fixed codebook, 3 <sup>rd</sup> subframe
130-139	Index for fixed codebook, 4 <sup>th</sup> subframe

Bit ordering of output bits from source encoder (11 bits/s).

Bits	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19-24	Index of 4 <sup>th</sup> LSF stage
25-31	Index of fixed and adaptive codebook gains, 1 <sup>st</sup> subframe
32-38	Index of fixed and adaptive codebook gains, 2 <sup>nd</sup> subframe
39-45	Index of fixed and adaptive codebook gains, 3 <sup>rd</sup> subframe
46-52	Index of fixed and adaptive codebook gains, 4 <sup>th</sup> subframe
53-60	Index of adaptive codebook, 1 <sup>st</sup> subframe
61-68	Index of adaptive codebook, 2 <sup>nd</sup> subframe
69-73	Index of adaptive codebook (relative), 3 <sup>rd</sup> subframe
74-78	Index of adaptive codebook (relative), 3 <sup>rd</sup> subframe
79-80	Index for LSF interpolation
81-100	Index for fixed codebook, 1 <sup>st</sup> subframe
101-120	Index for fixed codebook, 2 <sup>nd</sup> subframe
121-140	Index for fixed codebook, 3 <sup>rd</sup> subframe
141-160	Index for fixed codebook, 4 <sup>th</sup> subframe

Bit ordering of output bits from source encoder (6.65 bits/s).

Bits	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19-24	Index of 4 <sup>th</sup> LSF stage
25-31	Index of fixed and adaptive codebook gains, 1 <sup>st</sup> subframe
32-38	Index of fixed and adaptive codebook gains, 2 <sup>nd</sup> subframe
39-45	Index of fixed and adaptive codebook gains, 3 <sup>rd</sup> subframe
46-52	Index of fixed and adaptive codebook gains, 4 <sup>th</sup> subframe
53	Index for mode (LTP or PP)
LTP mode	
54-61	Index of adaptive codebook, 1 <sup>st</sup> subframe
62-69	Index of adaptive codebook, 2 <sup>nd</sup> subframe
70-74	Index of adaptive codebook (relative), 2 <sup>nd</sup> subframe
75-79	Index of adaptive codebook (relative), 4 <sup>th</sup> subframe
80-81	Index for LSF interpolation
82-94	Index for fixed codebook, 1 <sup>st</sup> subframe
95-107	Index for fixed codebook, 2 <sup>nd</sup> subframe
108-120	Index for fixed codebook, 3 <sup>rd</sup> subframe
121-133	Index for fixed codebook, 4 <sup>th</sup> subframe
PP mode	
	Index of pitch
	Index for LSF interpolation
	Index for fixed codebook, 1 <sup>st</sup> subframe
	Index for fixed codebook, 2 <sup>nd</sup> subframe
	Index for fixed codebook, 3 <sup>rd</sup> subframe
	Index for fixed codebook, 4 <sup>th</sup> subframe

Bit ordering of output bits from source encoder (5.8 bits/s).

Bits	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19-24	Index of 4 <sup>th</sup> LSF stage
25-31	Index of fixed and adaptive codebook gains, 1 <sup>st</sup> subframe
32-38	Index of fixed and adaptive codebook gains, 2 <sup>nd</sup> subframe
39-45	Index of fixed and adaptive codebook gains, 3 <sup>rd</sup> subframe
46-52	Index of fixed and adaptive codebook gains, 4 <sup>th</sup> subframe
53-60	Index of pitch
61-74	Index for fixed codebook, 1 <sup>st</sup> subframe
75-88	Index for fixed codebook, 2 <sup>nd</sup> subframe
89-102	Index for fixed codebook, 3 <sup>rd</sup> subframe
103-116	Index for fixed codebook, 4 <sup>th</sup> subframe

Bit ordering of output bits from source encoder (4.55 bits/s).

Bits	Description
1-6	Index of 1 <sup>st</sup> LSF stage
7-12	Index of 2 <sup>nd</sup> LSF stage
13-18	Index of 3 <sup>rd</sup> LSF stage
19	Index of pitch
20-25	Index of fixed and adaptive codebook gains, 1 <sup>st</sup> subframe
26-31	Index of fixed and adaptive codebook gains, 2 <sup>nd</sup> subframe
32-37	Index of fixed and adaptive codebook gains, 3 <sup>rd</sup> subframe
38-43	Index of fixed and adaptive codebook gains, 4 <sup>th</sup> subframe
44-51	Index of pitch
52-61	Index for fixed codebook, 1 <sup>st</sup> subframe
62-71	Index for fixed codebook, 2 <sup>nd</sup> subframe
72-81	Index for fixed codebook, 3 <sup>rd</sup> subframe
82-91	Index for fixed codebook, 4 <sup>th</sup> subframe

# APPENDIX C

## Bit ordering (channel coding)

Ordering of bits according to subjective importance (11 b/w/ R/TCH).

Bit, see table XXX	Description
1	b1-0
2	b1-1
3	b1-2
4	b1-3
5	b1-4
6	b1-5
7	b2-0
8	b2-1
9	b2-2
10	b2-3
11	b2-4
12	b2-5
13	p1ch1-0
14	p1ch1-1
15	p1ch1-2
16	p1ch1-3
17	p1ch1-4
18	p1ch1-5
19	p1ch1-6
20	p1ch1-7
21	p1ch1-8
22	p1ch1-9
23	p1ch1-10
24	p1ch1-11
25	p1ch1-12
26	p1ch1-13
27	p1ch1-14
28	p1ch1-15
29	p1-0
30	p1-1
31	p2-0
32	p2-1
33	p2-2
34	p2-3
35	p2-4
36	p2-5
37	p2-6
38	p2-7
39	p2-8
40	p2-9
41	p2-10
42	p2-11
43	p2-12
44	p2-13
45	p2-14
46	p2-15
47	p2-16
48	p2-17
49	p2-18
50	p2-19
51	p2-20
52	p2-21
53	p2-22
54	p2-23
55	p2-24
56	p2-25
57	p2-26
58	p2-27
59	p2-28
60	p2-29
61	p2-30
62	p2-31
63	p2-32
64	p2-33
65	p2-34
66	p2-35
67	p2-36
68	p2-37
69	p2-38
70	p2-39
71	p2-40
72	p2-41
73	p2-42
74	p2-43
75	p2-44
76	p2-45
77	p2-46
78	p2-47
79	p2-48
80	p2-49
81	p2-50
82	p2-51
83	p2-52
84	p2-53
85	p2-54
86	p2-55
87	p2-56
88	p2-57

89	p2ch4-0
90	p2ch4-1
91	p2ch4-2
92	p2ch4-3
93	p2ch4-4
94	p2ch4-5
95	b3-0
96	b3-1
97	b3-2
98	b3-3
99	b3-4
100	b3-5
101	b3-6
102	b3-7
103	b3-8
104	b3-9
105	b3-10
106	b3-11
107	b3-12
108	b3-13
109	b3-14
110	b3-15
111	b3-16
112	b3-17
113	b3-18
114	b3-19
115	b3-20
116	b3-21
117	b3-22
118	b3-23
119	b3-24
120	b3-25
121	b3-26
122	b3-27
123	b3-28
124	b3-29
125	b3-30
126	b3-31
127	b3-32
128	b3-33
129	b3-34



130	enc2-2
131	enc2-3
132	enc2-4
133	enc2-5
134	enc2-6
135	enc2-7
136	enc2-8
137	enc2-9
138	enc2-10
139	enc2-11
140	enc2-12
141	enc2-13
142	enc2-14
143	enc2-15
144	enc2-16
145	enc2-17
146	enc2-18
147	enc2-19
148	enc2-20
149	enc2-21
150	enc2-22
151	enc2-23
152	enc2-24
153	enc2-25
154	enc2-26
155	enc2-27
156	enc2-28
157	enc2-29
158	enc2-30
159	enc3-1
160	enc3-2
161	enc3-3
162	enc3-4
163	enc3-5
164	enc3-6
165	enc3-7
166	enc3-8
167	enc3-9
168	enc3-10
169	enc3-11
170	enc3-12
171	enc3-13
172	enc3-14
173	enc3-15
174	enc3-16
175	enc3-17
176	enc3-18
177	enc3-19
178	enc3-20
179	enc3-21
180	enc3-22
181	enc3-23
182	enc3-24
183	enc3-25
184	enc3-26
185	enc3-27
186	enc3-28
187	enc3-29
188	enc3-30
189	enc4-1
190	enc4-2
191	enc4-3
192	enc4-4
193	enc4-5
194	enc4-6
195	enc4-7
196	enc4-8
197	enc4-9
198	enc4-10

199	enc4-9
200	enc4-10
201	enc4-11
202	enc4-12
203	enc4-13
204	enc4-14
205	enc4-15
206	enc4-16
207	enc4-17
208	enc4-18
209	enc4-19
210	enc4-20
211	enc4-21
212	enc4-22
213	enc4-23
214	enc4-24
215	enc4-25
216	enc4-26
217	enc4-27
218	enc4-28
219	enc4-29
220	enc4-30
221	enc4-31
222	enc4-32
223	enc4-33
224	enc4-34
225	enc4-35
226	enc4-36
227	enc4-37
228	enc4-38
229	enc4-39
230	enc4-40
231	enc4-41
232	enc4-42
233	enc4-43
234	enc4-44
235	enc4-45
236	enc4-46
237	enc4-47
238	enc4-48
239	enc4-49
240	enc4-50
241	enc4-51
242	enc4-52
243	enc4-53
244	enc4-54
245	enc4-55
246	enc4-56
247	enc4-57
248	enc4-58
249	enc4-59
250	enc4-60
251	enc4-61
252	enc4-62
253	enc4-63
254	enc4-64
255	enc4-65
256	enc4-66
257	enc4-67
258	enc4-68
259	enc4-69
260	enc4-70
261	enc4-71
262	enc4-72
263	enc4-73
264	enc4-74
265	enc4-75
266	enc4-76
267	enc4-77
268	enc4-78
269	enc4-79
270	enc4-80
271	enc4-81
272	enc4-82
273	enc4-83
274	enc4-84
275	enc4-85
276	enc4-86
277	enc4-87
278	enc4-88
279	enc4-89
280	enc4-90
281	enc4-91
282	enc4-92
283	enc4-93
284	enc4-94
285	enc4-95
286	enc4-96
287	enc4-97
288	enc4-98
289	enc4-99
290	enc4-100

Ordering of bits according to subjective importance (4.0 bits FR TCH).

Bit	see table XXX	Description
1		bit 1-0
2		bit 1-1
3		bit 1-2
4		bit 1-3
5		bit 1-4
6		bit 1-5
7		bit 2-0
8		bit 2-1
9		bit 2-2
10		bit 2-3
11		bit 2-4
12		bit 2-5
13		bit 3-0
14		bit 3-1
15		bit 3-2
16		bit 3-3
17		bit 3-4
18		bit 3-5
19		bit 4-0
20		bit 4-1
21		bit 4-2
22		bit 4-3
23		bit 4-4
24		bit 4-5
25		bit 5-0
26		bit 5-1
27		bit 5-2
28		bit 5-3
29		bit 5-4
30		bit 5-5
31		bit 6-0
32		bit 6-1
33		bit 6-2
34		bit 6-3
35		bit 6-4
36		bit 6-5
37		bit 7-0
38		bit 7-1
39		bit 7-2
40		bit 7-3
41		bit 7-4
42		bit 7-5
43		bit 8-0
44		bit 8-1
45		bit 8-2
46		bit 8-3
47		bit 8-4
48		bit 8-5
49		bit 9-0
50		bit 9-1
51		bit 9-2
52		bit 9-3
53		bit 9-4
54		bit 9-5
55		bit 10-0
56		bit 10-1
57		bit 10-2
58		bit 10-3
59		bit 10-4
60		bit 10-5
61		bit 11-0
62		bit 11-1
63		bit 11-2
64		bit 11-3
65		bit 11-4
66		bit 11-5
67		bit 12-0
68		bit 12-1
69		bit 12-2
70		bit 12-3
71		bit 12-4
72		bit 12-5
73		bit 13-0
74		bit 13-1
75		bit 13-2
76		bit 13-3
77		bit 13-4
78		bit 13-5
79		bit 14-0
80		bit 14-1
81		bit 14-2
82		bit 14-3
83		bit 14-4
84		bit 14-5
85		bit 15-0
86		bit 15-1
87		bit 15-2
88		bit 15-3
89		bit 15-4
90		bit 15-5
91		bit 16-0
92		bit 16-1
93		bit 16-2
94		bit 16-3
95		bit 16-4
96		bit 16-5
97		bit 17-0
98		bit 17-1
99		bit 17-2
100		bit 17-3
101		bit 17-4
102		bit 17-5
103		bit 18-0
104		bit 18-1
105		bit 18-2
106		bit 18-3
107		bit 18-4
108		bit 18-5
109		bit 19-0
110		bit 19-1
111		bit 19-2
112		bit 19-3
113		bit 19-4
114		bit 19-5
115		bit 20-0
116		bit 20-1
117		bit 20-2
118		bit 20-3
119		bit 20-4
120		bit 20-5
121		bit 21-0
122		bit 21-1
123		bit 21-2
124		bit 21-3
125		bit 21-4
126		bit 21-5
127		bit 22-0

72	bit 22-1	bit 22-2
73	bit 22-3	bit 22-4
74	bit 22-5	bit 23-0
75	bit 23-1	bit 23-2
76	bit 23-3	bit 23-4
77	bit 23-5	bit 23-6
78	bit 23-7	bit 23-8
79	bit 23-9	bit 23-10
80	bit 23-11	bit 23-12
81	bit 23-13	bit 23-14
82	bit 23-15	bit 23-16
83	bit 23-17	bit 23-18
84	bit 23-19	bit 23-20
85	bit 23-21	bit 23-22
86	bit 23-23	bit 23-24
87	bit 23-25	bit 23-26
88	bit 23-27	bit 23-28
89	bit 23-29	bit 23-30
90	bit 23-31	bit 23-32
91	bit 23-33	bit 23-34
92	bit 23-35	bit 23-36
93	bit 23-37	bit 23-38
94	bit 23-39	bit 23-40
95	bit 23-41	bit 23-42
96	bit 23-43	bit 23-44
97	bit 23-45	bit 23-46
98	bit 23-47	bit 23-48
99	bit 23-49	bit 23-50
100	bit 23-51	bit 23-52
101	bit 23-53	bit 23-54
102	bit 23-55	bit 23-56
103	bit 23-57	bit 23-58
104	bit 23-59	bit 23-60
105	bit 23-61	bit 23-62
106	bit 23-63	bit 23-64
107	bit 23-65	bit 23-66
108	bit 23-67	bit 23-68
109	bit 23-69	bit 23-70
110	bit 23-71	bit 23-72
111	bit 23-73	bit 23-74
112	bit 23-75	bit 23-76
113	bit 23-77	bit 23-78
114	bit 23-79	bit 23-80
115	bit 23-81	bit 23-82
116	bit 23-83	bit 23-84
117	bit 23-85	bit 23-86
118	bit 23-87	bit 23-88
119	bit 23-89	bit 23-90
120	bit 23-91	bit 23-92
121	bit 23-93	bit 23-94
122	bit 23-95	bit 23-96
123	bit 23-97	bit 23-98
124	bit 23-99	bit 23-100
125	bit 23-101	bit 23-102
126	bit 23-103	bit 23-104
127	bit 23-105	bit 23-106

128	exc3-7
129	exc3-8
130	exc3-9
131	exc3-10
132	exc3-11
133	exc3-12
134	exc3-13
135	exc3-14
136	exc3-15
137	exc3-16
138	exc3-17
139	exc3-18
140	exc3-19
141	exc4-0
142	exc4-1
143	exc4-2
144	exc4-3
145	exc4-4
146	exc4-5
147	exc4-6
148	exc4-7
149	exc4-8
150	exc4-9
151	exc4-10
152	exc4-11
153	exc4-12
154	exc4-13
155	exc4-14
156	exc4-15
157	exc4-16
158	exc4-17
159	exc4-18
160	exc4-19

Ordering of bits according to subjective importance (6.65 kb/s FR-TCH).

Bit. use table XXX	Description
54	pitch-0
55	pitch-1
56	pitch-2
57	pitch-3
58	pitch-4
59	pitch-5
1	br1-0
2	br1-1
3	br1-2
4	br1-3
5	br1-4
6	br1-5
25	pitch-0
26	pitch-1
27	pitch-2
28	pitch-3
32	pitch-0
33	pitch-1
34	pitch-2
35	pitch-3
39	pitch-0
40	pitch-1
41	pitch-2
42	pitch-3
46	pitch-0
47	pitch-1
48	pitch-2
49	pitch-3
29	pitch-4
36	pitch-4
43	pitch-4
50	pitch-4
51	pitch-0
58	exc1-0 pitch-0 (Third subframe)
59	exc1-1 pitch-1 (Second subframe)
7	br2-0
8	br2-1
9	br2-2
10	br2-3
11	br2-4
12	br2-5
30	pitch-3
37	pitch-3
44	pitch-3
31	pitch-3
62	exc1-0 pitch-0 (Third subframe)
63	exc1-1 pitch-1 (Third subframe)
64	exc1-2 pitch-2 (Third subframe)
65	exc1-3 pitch-3 (Third subframe)
66	exc1-4 pitch-4 (Third subframe)
80	exc2-0 pitch-0 (Third subframe)
100	exc2-1 pitch-1 (Second subframe)
116	exc3-0 pitch-0 (Fourth subframe)
117	exc3-1 pitch-1 (Fourth subframe)
118	exc3-2 pitch-2 (Fourth subframe)
13	br3-0
14	br3-1
15	br3-2
16	br3-3
17	br3-4
18	br3-5
19	br4-0
20	br4-1

31	h16-2
32	h16-1
33	ec1-3 ec1(hp)
34	ec1-4 ec1(hp)
35	ec1-5 ec1(hp)
36	ec1-6 ec1(hp)
37	ec1-7 ec1(hp)
38	ec1-8 ec1(hp)
39	ec1-9 ec1(hp)
40	ec1-10 ec1(hp)
41	ec2-1 ec2(hp)
42	ec2-2 ec2(hp)
43	ec2-3 ec2(hp)
44	ec2-4 ec2(hp)
45	ec2-5 ec2(hp)
46	ec2-6 ec2(hp)
47	ec2-7 ec2(hp)
48	ec2-8 ec2(hp)
49	ec2-9 ec2(hp)
50	ec2-10 ec2(hp)
51	ec3-1 ec3(hp)
52	ec3-2 ec3(hp)
53	ec3-3 ec3(hp)
54	ec3-4 ec3(hp)
55	ec3-5 ec3(hp)
56	ec3-6 ec3(hp)
57	ec3-7 ec3(hp)
58	ec3-8 ec3(hp)
59	ec3-9 ec3(hp)
60	ec3-10 ec3(hp)
61	ec4-1 ec4(hp)
62	ec4-2 ec4(hp)
63	ec4-3 ec4(hp)
64	ec4-4 ec4(hp)
65	ec4-5 ec4(hp)
66	ec4-6 ec4(hp)
67	ec4-7 ec4(hp)
68	ec4-8 ec4(hp)
69	ec4-9 ec4(hp)
70	ec4-10 ec4(hp)
71	ec5-1 ec5(hp)
72	ec5-2 ec5(hp)
73	ec5-3 ec5(hp)
74	ec5-4 ec5(hp)
75	ec5-5 ec5(hp)
76	ec5-6 ec5(hp)
77	ec5-7 ec5(hp)
78	ec5-8 ec5(hp)
79	ec5-9 ec5(hp)
80	ec5-10 ec5(hp)
81	ec6-1 ec6(hp)
82	ec6-2 ec6(hp)
83	ec6-3 ec6(hp)
84	ec6-4 ec6(hp)
85	ec6-5 ec6(hp)
86	ec6-6 ec6(hp)
87	ec6-7 ec6(hp)
88	ec6-8 ec6(hp)
89	ec6-9 ec6(hp)
90	ec6-10 ec6(hp)
91	ec7-1 ec7(hp)
92	ec7-2 ec7(hp)
93	ec7-3 ec7(hp)
94	ec7-4 ec7(hp)
95	ec7-5 ec7(hp)
96	ec7-6 ec7(hp)
97	ec7-7 ec7(hp)
98	ec7-8 ec7(hp)
99	ec7-9 ec7(hp)
100	ec7-10 ec7(hp)
101	ec8-1 ec8(hp)
102	ec8-2 ec8(hp)
103	ec8-3 ec8(hp)
104	ec8-4 ec8(hp)
105	ec8-5 ec8(hp)
106	ec8-6 ec8(hp)
107	ec8-7 ec8(hp)
108	ec8-8 ec8(hp)
109	ec8-9 ec8(hp)
110	ec8-10 ec8(hp)
111	ec9-1 ec9(hp)
112	ec9-2 ec9(hp)
113	ec9-3 ec9(hp)
114	ec9-4 ec9(hp)
115	ec9-5 ec9(hp)
116	ec9-6 ec9(hp)
117	ec9-7 ec9(hp)
118	ec9-8 ec9(hp)
119	ec9-9 ec9(hp)
120	ec9-10 ec9(hp)
121	ec10-1 ec10(hp)
122	ec10-2 ec10(hp)
123	ec10-3 ec10(hp)
124	ec10-4 ec10(hp)
125	ec10-5 ec10(hp)
126	ec10-6 ec10(hp)
127	ec10-7 ec10(hp)
128	ec10-8 ec10(hp)
129	ec10-9 ec10(hp)
130	ec10-10 ec10(hp)
131	ec11-1 ec11(hp)
132	ec11-2 ec11(hp)
133	ec11-3 ec11(hp)
134	ec11-4 ec11(hp)
135	ec11-5 ec11(hp)
136	ec11-6 ec11(hp)
137	ec11-7 ec11(hp)
138	ec11-8 ec11(hp)
139	ec11-9 ec11(hp)
140	ec11-10 ec11(hp)
141	ec12-1 ec12(hp)
142	ec12-2 ec12(hp)
143	ec12-3 ec12(hp)
144	ec12-4 ec12(hp)
145	ec12-5 ec12(hp)
146	ec12-6 ec12(hp)
147	ec12-7 ec12(hp)
148	ec12-8 ec12(hp)
149	ec12-9 ec12(hp)
150	ec12-10 ec12(hp)
151	ec13-1 ec13(hp)
152	ec13-2 ec13(hp)
153	ec13-3 ec13(hp)
154	ec13-4 ec13(hp)
155	ec13-5 ec13(hp)
156	ec13-6 ec13(hp)
157	ec13-7 ec13(hp)
158	ec13-8 ec13(hp)
159	ec13-9 ec13(hp)
160	ec13-10 ec13(hp)
161	ec14-1 ec14(hp)
162	ec14-2 ec14(hp)
163	ec14-3 ec14(hp)
164	ec14-4 ec14(hp)
165	ec14-5 ec14(hp)
166	ec14-6 ec14(hp)
167	ec14-7 ec14(hp)
168	ec14-8 ec14(hp)
169	ec14-9 ec14(hp)
170	ec14-10 ec14(hp)
171	ec15-1 ec15(hp)
172	ec15-2 ec15(hp)
173	ec15-3 ec15(hp)
174	ec15-4 ec15(hp)
175	ec15-5 ec15(hp)
176	ec15-6 ec15(hp)
177	ec15-7 ec15(hp)
178	ec15-8 ec15(hp)
179	ec15-9 ec15(hp)
180	ec15-10 ec15(hp)
181	ec16-1 ec16(hp)
182	ec16-2 ec16(hp)
183	ec16-3 ec16(hp)
184	ec16-4 ec16(hp)
185	ec16-5 ec16(hp)
186	ec16-6 ec16(hp)
187	ec16-7 ec16(hp)
188	ec16-8 ec16(hp)
189	ec16-9 ec16(hp)
190	ec16-10 ec16(hp)
191	ec17-1 ec17(hp)
192	ec17-2 ec17(hp)
193	ec17-3 ec17(hp)
194	ec17-4 ec17(hp)
195	ec17-5 ec17(hp)
196	ec17-6 ec17(hp)
197	ec17-7 ec17(hp)
198	ec17-8 ec17(hp)
199	ec17-9 ec17(hp)
200	ec17-10 ec17(hp)

112	ec16-16
113	ec16-17
114	ec16-18
115	ec16-19
116	ec16-20
117	ec16-21
118	ec16-22
119	ec16-23
120	ec16-24
121	ec16-25
122	ec16-26
123	ec16-27
124	ec16-28
125	ec16-29
126	ec16-30
127	ec16-31
128	ec16-32
129	ec16-33
130	ec16-34
131	ec16-35
132	ec16-36
133	ec16-37
134	ec16-38
135	ec16-39
136	ec16-40
137	ec16-41
138	ec16-42
139	ec16-43
140	ec16-44
141	ec16-45
142	ec16-46
143	ec16-47
144	ec16-48
145	ec16-49
146	ec16-50
147	ec16-51
148	ec16-52
149	ec16-53
150	ec16-54
151	ec16-55
152	ec16-56
153	ec16-57
154	ec16-58
155	ec16-59
156	ec16-60
157	ec16-61
158	ec16-62
159	ec16-63
160	ec16-64
161	ec16-65
162	ec16-66
163	ec16-67
164	ec16-68
165	ec16-69
166	ec16-70
167	ec16-71
168	ec16-72
169	ec16-73
170	ec16-74
171	ec16-75
172	ec16-76
173	ec16-77
174	ec16-78
175	ec16-79
176	ec16-80
177	ec16-81
178	ec16-82
179	ec16-83
180	ec16-84
181	ec16-85
182	ec16-86
183	ec16-87
184	ec16-88
185	ec16-89
186	ec16-90
187	ec16-91
188	ec16-92
189	ec16-93
190	ec16-94
191	ec16-95
192	ec16-96
193	ec16-97
194	ec16-98
195	ec16-99
196	ec16-100

Order of bits according to subjective importance (5.8 better FITCH).

Bit	ec16-XXX	Description
1	ec16-1	ec16-1
2	ec16-2	ec16-2
3	ec16-3	ec16-3
4	ec16-4	ec16-4
5	ec16-5	ec16-5
6	ec16-6	ec16-6
7	ec16-7	ec16-7
8	ec16-8	ec16-8
9	ec16-9	ec16-9
10	ec16-10	ec16-10
11	ec16-11	ec16-11
12	ec16-12	ec16-12
13	ec16-13	ec16-13
14	ec16-14	ec16-14
15	ec16-15	ec16-15
16	ec16-16	ec16-16
17	ec16-17	ec16-17
18	ec16-18	ec16-18
19	ec16-19	ec16-19
20	ec16-20	ec16-20
21	ec16-21	ec16-21
22	ec16-22	ec16-22
23	ec16-23	ec16-23
24	ec16-24	ec16-24

31	rule-6
32	rule-6
33	rule-6
34	rule-6
35	rule-6
36	rule-6
37	rule-6
38	rule-6
39	rule-6
40	rule-6
41	rule-6
42	rule-6
43	rule-6
44	rule-6
45	rule-6
46	rule-6
47	rule-6
48	rule-6
49	rule-6
50	rule-6
51	rule-6
52	rule-6
53	rule-6
54	rule-6
55	rule-6
56	rule-6
57	rule-6
58	rule-6
59	rule-6
60	rule-6
61	rule-6
62	rule-6
63	rule-6
64	rule-6
65	rule-6
66	rule-6
67	rule-6
68	rule-6
69	rule-6
70	rule-6
71	rule-6
72	rule-6
73	rule-6
74	rule-6
75	rule-6
76	rule-6
77	rule-6
78	rule-6
79	rule-6
80	rule-6
81	rule-6
82	rule-6
83	rule-6
84	rule-6
85	rule-6
86	rule-6
87	rule-6
88	rule-6
89	rule-6
90	rule-6
91	rule-6
92	rule-6
93	rule-6
94	rule-6
95	rule-6
96	rule-6
97	rule-6
98	rule-6
99	rule-6
100	rule-6
101	rule-6
102	rule-6
103	rule-6
104	rule-6
105	rule-6
106	rule-6
107	rule-6
108	rule-6
109	rule-6
110	rule-6
111	rule-6
112	rule-6
113	rule-6
114	rule-6
115	rule-6
116	rule-6

Ordering of bits according to subjective importance (8.0 kHz HRTN).

Bits, see table XXX	Description
1	rule-0
2	rule-1
3	rule-2
4	rule-3
5	rule-4
6	rule-5
7	rule-6
8	rule-7
9	rule-8
10	rule-9
11	rule-10
12	rule-11
13	rule-12
14	rule-13
15	rule-14
16	rule-15
17	rule-16
18	rule-17
19	rule-18
20	rule-19
21	rule-20
22	rule-21
23	rule-22
24	rule-23
25	rule-24
26	rule-25
27	rule-26
28	rule-27
29	rule-28
30	rule-29
31	rule-30
32	rule-31
33	rule-32
34	rule-33
35	rule-34
36	rule-35
37	rule-36
38	rule-37
39	rule-38
40	rule-39
41	rule-40
42	rule-41
43	rule-42
44	rule-43
45	rule-44
46	rule-45
47	rule-46
48	rule-47
49	rule-48
50	rule-49
51	rule-50
52	rule-51
53	rule-52
54	rule-53
55	rule-54
56	rule-55
57	rule-56
58	rule-57
59	rule-58
60	rule-59
61	rule-60
62	rule-61
63	rule-62
64	rule-63
65	rule-64
66	rule-65
67	rule-66
68	rule-67
69	rule-68
70	rule-69
71	rule-70
72	rule-71
73	rule-72
74	rule-73
75	rule-74
76	rule-75
77	rule-76
78	rule-77
79	rule-78
80	rule-79
81	rule-80
82	rule-81
83	rule-82
84	rule-83
85	rule-84
86	rule-85
87	rule-86
88	rule-87
89	rule-88
90	rule-89
91	rule-90
92	rule-91
93	rule-92
94	rule-93
95	rule-94
96	rule-95
97	rule-96
98	rule-97
99	rule-98
100	rule-99
101	rule-100
102	rule-101
103	rule-102
104	rule-103
105	rule-104
106	rule-105
107	rule-106
108	rule-107
109	rule-108
110	rule-109
111	rule-110
112	rule-111
113	rule-112
114	rule-113
115	rule-114
116	rule-115

24		h2c4
25		h2c5
26		h2c6
27		h2c7
28		h2c8
29		h2c9
30		h2c10
31		h2c11
32		h2c12
33		h2c13
34		h2c14
35		h2c15
36		h2c16
37		h2c17
38		h2c18
39		h2c19
40		h2c20
41		h2c21
42		h2c22
43		h2c23
44		h2c24
45		h2c25
46		h2c26
47		h2c27
48		h2c28
49		h2c29
50		h2c30
51		h2c31
52		h2c32
53		h2c33
54		h2c34
55		h2c35
56		h2c36
57		h2c37
58		h2c38
59		h2c39
60		h2c40
61		h2c41
62		h2c42
63		h2c43
64		h2c44
65		h2c45
66		h2c46
67		h2c47
68		h2c48
69		h2c49
70		h2c50
71		h2c51
72		h2c52
73		h2c53
74		h2c54
75		h2c55
76		h2c56
77		h2c57
78		h2c58
79		h2c59
80		h2c60
81		h2c61
82		h2c62
83		h2c63
84		h2c64
85		h2c65
86		h2c66
87		h2c67
88		h2c68
89		h2c69
90		h2c70
91		h2c71
92		h2c72
93		h2c73
94		h2c74
95		h2c75
96		h2c76
97		h2c77
98		h2c78
99		h2c79
100		h2c80
101		h2c81
102		h2c82
103		h2c83
104		h2c84
105		h2c85
106		h2c86
107		h2c87
108		h2c88
109		h2c89
110		h2c90
111		h2c91
112		h2c92
113		h2c93
114		h2c94
115		h2c95
116		h2c96
117		h2c97
118		h2c98
119		h2c99
120		h2c100
121		h2c101
122		h2c102
123		h2c103
124		h2c104
125		h2c105
126		h2c106
127		h2c107
128		h2c108

129		h2c109
130		h2c110
131		h2c111
132		h2c112
133		h2c113
134		h2c114
135		h2c115
136		h2c116
137		h2c117
138		h2c118
139		h2c119
140		h2c120
141		h2c121
142		h2c122
143		h2c123
144		h2c124
145		h2c125
146		h2c126
147		h2c127
148		h2c128
149		h2c129
150		h2c130
151		h2c131
152		h2c132
153		h2c133
154		h2c134
155		h2c135
156		h2c136
157		h2c137
158		h2c138
159		h2c139
160		h2c140

Ordering of bits according to subjective importance (6.65 bina HRTN).

Bit, see table XXX	Description
31	mode-0
34	pitch-0
35	pitch-1
36	pitch-2
37	pitch-3
38	pitch-4
39	pitch-5
40	pitch-6
41	pitch-7
42	pitch-8
43	pitch-9
44	pitch-10
45	pitch-11
46	pitch-12
47	pitch-13
48	pitch-14
49	pitch-15
50	pitch-16
51	pitch-17
52	pitch-18
53	pitch-19
54	pitch-20
55	pitch-21
56	pitch-22
57	pitch-23
58	pitch-24
59	pitch-25
60	pitch-26
61	pitch-27
62	pitch-28
63	pitch-29
64	pitch-30
65	pitch-31
66	pitch-32
67	pitch-33
68	pitch-34
69	pitch-35
70	pitch-36
71	pitch-37
72	pitch-38
73	pitch-39
74	pitch-40
75	pitch-41
76	pitch-42
77	pitch-43
78	pitch-44
79	pitch-45
80	pitch-46
81	pitch-47
82	pitch-48
83	pitch-49
84	pitch-50
85	pitch-51
86	pitch-52
87	pitch-53
88	pitch-54
89	pitch-55
90	pitch-56
91	pitch-57
92	pitch-58
93	pitch-59
94	pitch-60
95	pitch-61
96	pitch-62
97	pitch-63
98	pitch-64
99	pitch-65
100	pitch-66
101	pitch-67
102	pitch-68
103	pitch-69
104	pitch-70
105	pitch-71
106	pitch-72
107	pitch-73
108	pitch-74
109	pitch-75
110	pitch-76
111	pitch-77
112	pitch-78
113	pitch-79
114	pitch-80
115	pitch-81
116	pitch-82
117	pitch-83
118	pitch-84
119	pitch-85
120	pitch-86
121	pitch-87
122	pitch-88
123	pitch-89
124	pitch-90
125	pitch-91
126	pitch-92
127	pitch-93
128	pitch-94
129	pitch-95
130	pitch-96
131	pitch-97
132	pitch-98
133	pitch-99
134	pitch-100
135	pitch-101
136	pitch-102
137	pitch-103
138	pitch-104
139	pitch-105
140	pitch-106
141	pitch-107
142	pitch-108
143	pitch-109
144	pitch-110
145	pitch-111
146	pitch-112
147	pitch-113
148	pitch-114
149	pitch-115
150	pitch-116
151	pitch-117
152	pitch-118
153	pitch-119
154	pitch-120
155	pitch-121
156	pitch-122
157	pitch-123
158	pitch-124
159	pitch-125
160	pitch-126
161	pitch-127
162	pitch-128
163	pitch-129
164	pitch-130
165	pitch-131
166	pitch-132
167	pitch-133
168	pitch-134
169	pitch-135
170	pitch-136
171	pitch-137
172	pitch-138
173	pitch-139
174	pitch-140
175	pitch-141
176	pitch-142
177	pitch-143
178	pitch-144
179	pitch-145
180	pitch-146
181	pitch-147
182	pitch-148
183	pitch-149
184	pitch-150
185	pitch-151
186	pitch-152
187	pitch-153
188	pitch-154
189	pitch-155
190	pitch-156
191	pitch-157
192	pitch-158
193	pitch-159
194	pitch-160
195	pitch-161
196	pitch-162
197	pitch-163
198	pitch-164
199	pitch-165
200	pitch-166
201	pitch-167
202	pitch-168
203	pitch-169
204	pitch-170
205	pitch-171
206	pitch-172
207	pitch-173
208	pitch-174
209	pitch-175
210	pitch-176
211	pitch-177
212	pitch-178
213	pitch-179
214	pitch-180
215	pitch-181
216	pitch-182
217	pitch-183
218	pitch-184
219	pitch-185
220	pitch-186
221	pitch-187
222	pitch-188
223	pitch-189
224	pitch-190
225	pitch-191
226	pitch-192
227	pitch-193
228	pitch-194
229	pitch-195
230	pitch-196
231	pitch-197
232	pitch-198
233	pitch-199
234	pitch-200
235	pitch-201
236	pitch-202
237	pitch-203
238	pitch-204
239	pitch-205
240	pitch-206
241	pitch-207
242	pitch-208
243	pitch-209
244	pitch-210
245	pitch-211
246	pitch-212
247	pitch-213
248	pitch-214
249	pitch-215
250	pitch-216
251	pitch-217
252	pitch-218
253	pitch-219
254	pitch-220
255	pitch-221
256	pitch-222
257	pitch-223
258	pitch-224
259	pitch-225
260	pitch-226
261	pitch-227
262	pitch-228
263	pitch-229
264	pitch-230
265	pitch-231
266	pitch-232
267	pitch-233
268	pitch-234
269	pitch-235
270	pitch-236
271	pitch-237
272	pitch-238
273	pitch-239
274	pitch-240
275	pitch-241
276	pitch-242
277	pitch-243
278	pitch-244
279	pitch-245
280	pitch-246
281	pitch-247
282	pitch-248
283	pitch-249
284	pitch-250
285	pitch-251
286	pitch-252
287	pitch-253
288	pitch-254
289	pitch-255
290	pitch-256
291	pitch-257
292	pitch-258
293	pitch-259
294	pitch-260
295	pitch-261
296	pitch-262
297	pitch-263
298	pitch-264
299	pitch-265
300	pitch-266
301	pitch-267
302	pitch-268
303	pitch-269
304	pitch-270
305	pitch-271
306	pitch-272
307	pitch-273
308	pitch-274
309	pitch-275
310	pitch-276
311	pitch-277
312	pitch-278
313	pitch-279
314	pitch-280
315	pitch-281
316	pitch-282
317	pitch-283
318	pitch-284
319	pitch-285
320	pitch-286
321	pitch-287
322	pitch-288
323	pitch-289
324	pitch-290
325	pitch-291
326	pitch-292
327	pitch-293
328	pitch-294
329	pitch-295
330	pitch-296
331	pitch-297
332	pitch-298
333	pitch-299
334	pitch-300
335	pitch-301
336	pitch-302
337	pitch-303
338	pitch-304
339	pitch-305
340	pitch-306
341	pitch-307
342	pitch-308
343	pitch-309
344	pitch-310
345	pitch-311
346	pitch-312
347	pitch-313
348	pitch-314
349	pitch-315
350	pitch-316
351	pitch-317
352	pitch-318
353	pitch-319
354	pitch-320
355	pitch-321
356	pitch-322
357	pitch-323
358	pitch-324
359	pitch-325
360	pitch-326
361	pitch-327
362	pitch-328
363	pitch-329
364	pitch-330
365	pitch-331
366	pitch-332
367	pitch-333
368	pitch-334
369	pitch-335
370	pitch-336
371	pitch-337
372	pitch-338
373	pitch-339
374	pitch-340
375	pitch-341
376	pitch-342
377	pitch-343
378	pitch-344
379	pitch-345
380	pitch-346
381	pitch-347
382	pitch-348
383	pitch-349
384	pitch-350
385	pitch-351
386	pitch-352
387	pitch-353
388	pitch-354
389	pitch-355
390	pitch-356
391	pitch-357
392	pitch-358
393	pitch-359
394	pitch-360
395	pitch-361
396	pitch-362
397	pitch-363
398	pitch-364
399	pitch-365
400	pitch-366
401	pitch-367
402	pitch-368
403	pitch-369
404	pitch-370
405	pitch-371
406	pitch-372
407	pitch-373
408	pitch-374
409	pitch-375
410	pitch-376
411	pitch-377
412	pitch-378
413	pitch-379
414	pitch-380
415	pitch-381
416	pitch-382
417	pitch-383
418	pitch-384
419	pitch-385
420	pitch-386
421	pitch-387
422	pitch-388
423	pitch-389
424	pitch-390
425	pitch-391
426	pitch-392
427	pitch-393
428	pitch-394
429	pitch-395
430	pitch-396
431	pitch-397
432	pitch-398
433	pitch-399
434	pitch-400
435	pitch-401
436	pitch-402
437	pitch-403
438	pitch-404
439	pitch-405
440	pitch-406
441	pitch-407
442	pitch-408
443	pitch-409
444	pitch-410
445	pitch-411
446	pitch-412
447	pitch-413
448	pitch-414
449	pitch-415
450	pitch-416
451	pitch-417
452	pitch-418
453	pitch-419
454	pitch-420
455	pitch-421
456	pitch-422
457	pitch-423
458	pitch-424
459	pitch-425
460	pitch-426
461	pitch-427
462	pitch-428
463	pitch-429
464	pitch-430
465	pitch-431
466	pitch-432
467	pitch-433
468	pitch-434
469	pitch-435
470	pitch-436
471	pitch-437
472	pitch-438
473	pitch-439
474	pitch-440
475	pitch-441
476	pitch-442
477	pitch-443
478	pitch-444
479	pitch-445
480	pitch-446
481	pitch-447
482	pitch-448
483	pitch-449
484	pitch-450
485	pitch-451
486	pitch-452
487	pitch-453
488	pitch-454
489	pitch-455
490	pitch-456
491	pitch-457
492	pitch-458
493	pitch-459
494	pitch-460
495	pitch-461
496	pitch-462
497	pitch-463
498	pitch-464
499	pitch-465
500	pitch-466
501	pitch-467
502	pitch-468
503	pitch-469
504	pitch-470
505	pitch-471
506	pitch-472
507	pitch-473
508	pitch-474
509	pitch-475
510	pitch-476
511	pitch-477
512	pitch-478
513	pitch-479
514	pitch-480
515	pitch-481
516	pitch-482
517	pitch-483
518	pitch-484
519	pitch-485
520	pitch-486
521	pitch-487
522	pitch-488
523	pitch-489
524	pitch-490
525	pitch-491
526	pitch-492
527	pitch-493
528	pitch-494
529	pitch-495
530	pitch-496
531	pitch-497
532	pitch-498
533	pitch-499
534	pitch-500
535	pitch-501
536	pitch-502
537	pitch-503
538	pitch-504
539	pitch-505
540	pitch-506
541	pitch-507
542	pitch-508
543	pitch-509
544	pitch-510
545	pitch-511
546	pitch-512
547	pitch-513
548	pitch-514
549	pitch-515
550	pitch-516
551	pitch-517
552	pitch-518
553	pitch-519
554	pitch-520
555	pitch-521
556	pitch-522
557	pitch-523
558	pitch-524
559	pitch-525
560	pitch-526
561	pitch-527
562	pitch-528
563	pitch-529
564	pitch-530
565	pitch-531
566	

112	enc-16
79	enc-17
87	enc-17
113	enc-17
113	enc-17

Ordering of this according to subjective importance (5.8 below HATCH).

Enc. no. table XXX	Description
25	enc-10
26	enc-10
32	enc-10
33	enc-10
39	enc-10
40	enc-10
46	enc-10
47	enc-10
1	enc-10
2	enc-10
3	enc-10
4	enc-10
5	enc-10
6	enc-10
37	enc-10
34	enc-10
41	enc-10
48	enc-10
51	enc-10
52	enc-10
53	enc-10
54	enc-10
55	enc-10
56	enc-10
57	enc-10
58	enc-10
59	enc-10
60	enc-10
61	enc-10
62	enc-10
63	enc-10
64	enc-10
65	enc-10
66	enc-10
67	enc-10
68	enc-10
69	enc-10
70	enc-10
71	enc-10
72	enc-10
73	enc-10
74	enc-10
75	enc-10
76	enc-10
77	enc-10
78	enc-10
79	enc-10
80	enc-10
81	enc-10
82	enc-10
83	enc-10
84	enc-10
85	enc-10
86	enc-10
87	enc-10
88	enc-10
89	enc-10
90	enc-10
91	enc-10
92	enc-10
93	enc-10
94	enc-10
95	enc-10
96	enc-10
97	enc-10
98	enc-10
99	enc-10
100	enc-10
101	enc-10
102	enc-10
103	enc-10
104	enc-10
105	enc-10
106	enc-10
107	enc-10
108	enc-10
109	enc-10
110	enc-10
111	enc-10
112	enc-10
113	enc-10
114	enc-10
115	enc-10
116	enc-10

61	enc-10
62	enc-10
63	enc-10
64	enc-10
65	enc-10
66	enc-10
67	enc-10
68	enc-10
69	enc-10
70	enc-10
71	enc-10
72	enc-10
73	enc-10
74	enc-10
75	enc-10
76	enc-10
77	enc-10
78	enc-10
79	enc-10
80	enc-10
81	enc-10
82	enc-10
83	enc-10
84	enc-10
85	enc-10
86	enc-10
87	enc-10
88	enc-10
89	enc-10
90	enc-10
91	enc-10
92	enc-10
93	enc-10
94	enc-10
95	enc-10
96	enc-10
97	enc-10
98	enc-10
99	enc-10
100	enc-10
101	enc-10
102	enc-10
103	enc-10
104	enc-10
105	enc-10
106	enc-10
107	enc-10
108	enc-10
109	enc-10
110	enc-10
111	enc-10
112	enc-10
113	enc-10
114	enc-10
115	enc-10
116	enc-10



Ordering of bits according to subjective importance (4.53 bits/HRTCH).

Bits, see table XXX	Description
20	rela1-0
26	rela2-0
44	pitch-0
45	pitch-1
46	pitch-2
32	pitch-3
38	pitch-4
31	pitch-5
27	pitch-6
33	pitch-7
39	pitch-8
19	pitch-9
1	pitch-10
2	pitch-11
3	pitch-12
4	pitch-13
5	pitch-14
6	pitch-15
7	pitch-16
8	pitch-17
9	pitch-18
22	pitch-19
28	pitch-20
34	pitch-21
40	pitch-22
23	pitch-23
29	pitch-24
35	pitch-25
41	pitch-26
47	pitch-27
10	pitch-28
11	pitch-29
12	pitch-30
24	pitch-31
30	pitch-32
36	pitch-33
42	pitch-34
48	pitch-35
13	pitch-36
14	pitch-37
15	pitch-38
16	pitch-39
17	pitch-40
18	pitch-41
37	pitch-42
38	pitch-43
39	pitch-44
40	pitch-45
41	pitch-46
42	pitch-47
43	pitch-48
44	pitch-49
45	pitch-50
46	pitch-51
47	pitch-52
48	pitch-53
49	pitch-54
50	pitch-55
51	pitch-56
52	pitch-57
53	pitch-58
54	pitch-59
55	pitch-60
56	pitch-61
57	pitch-62
58	pitch-63
59	pitch-64
60	pitch-65
61	pitch-66
62	pitch-67
63	pitch-68
64	pitch-69
65	pitch-70
66	pitch-71

67	exc2-5
72	exc2-6
73	exc2-7
74	exc2-8
75	exc2-9
76	exc2-10
77	exc2-11
82	exc2-12
83	exc2-13
84	exc2-14
85	exc2-15
86	exc2-16
87	exc2-17
88	exc2-18
89	exc2-19
90	exc2-20
91	exc2-21

# CLAIMS

I claim:

1. A speech codec using long term preprocessing of a speech signal having a pitch lag, the speech codec comprising:  
an adaptive codebook;  
an encoder, coupled to the adaptive codebook, that estimates the pitch lag; and  
the encoder applying continuous warping of the speech signal using the estimated pitch lag.
2. The speech codec of claim 1 wherein the speech signal comprises a weighted speech signal.
3. The speech codec of any of claims 1 and 2 wherein the encoder searches for a best local delay using linear time weighting.
4. The speech codec of any of claims 1 and 2 wherein the continuous warping comprises translating the speech signal from a first time region to a second time region.
5. The speech codec of claim 1 wherein the speech signal comprises a residual signal.
6. A speech codec using long term preprocessing of a speech signal, the speech codec comprising:  
an adaptive codebook;

- an encoder, coupled to the adaptive codebook, that continuously warps the speech signal to a target contour; and  
the encoder searches for a best local delay using linear time weighting.
7. The speech codec of claim 6 wherein the speech signal comprises a weighted speech signal.
  8. The speech codec of claim 6 wherein the speech signal comprises a residual signal.
  9. The speech codec of claim 6 wherein the encoder processing circuit identifies a limited search range for the best local delay.
  10. The speech codec of claim 9 wherein the identification by the encoder of the limited search range is based at least in part on sharpness of the speech signal.
  11. The speech codec of claim 9 wherein the identification by the encoder of the limited search range is based at least in part on a classification of the speech signal.
  12. The speech codec of claim 11 wherein the classification of the speech signal involves classifying the speech signal as either voiced or unvoiced speech.
  13. The speech codec of claim 6 wherein the speech signal having a previous pitch lag and a current pitch lag, and the encoder utilizes estimates of the previous pitch lag and the current pitch lag to generate the target contour.

1/11

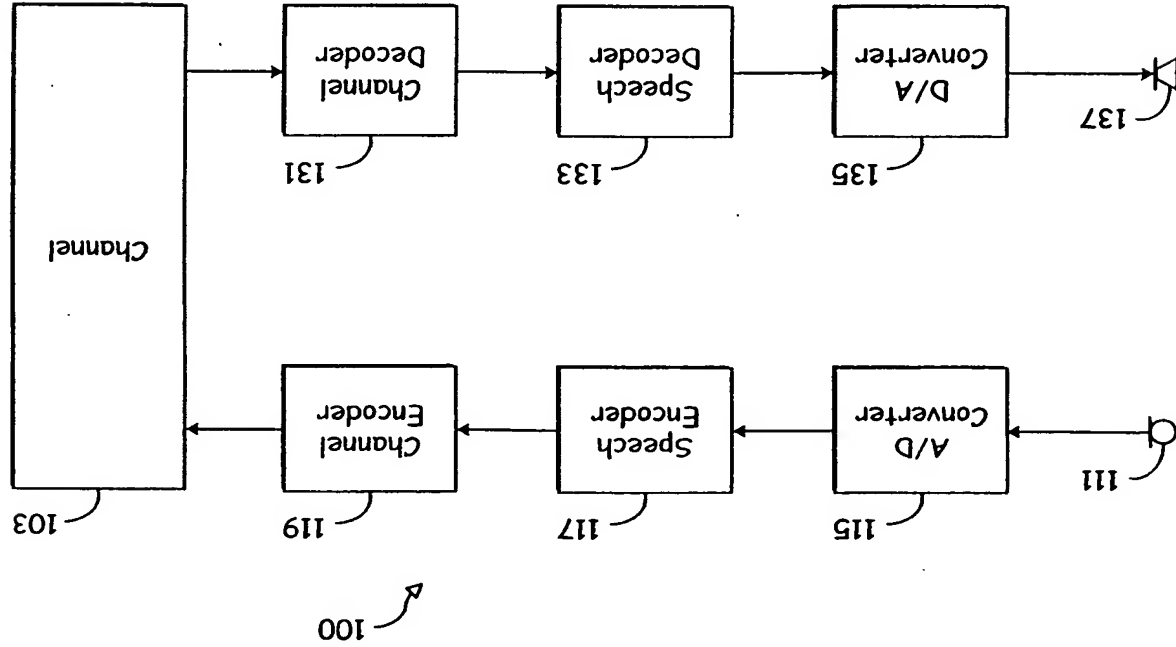


Fig. 1a

2/11

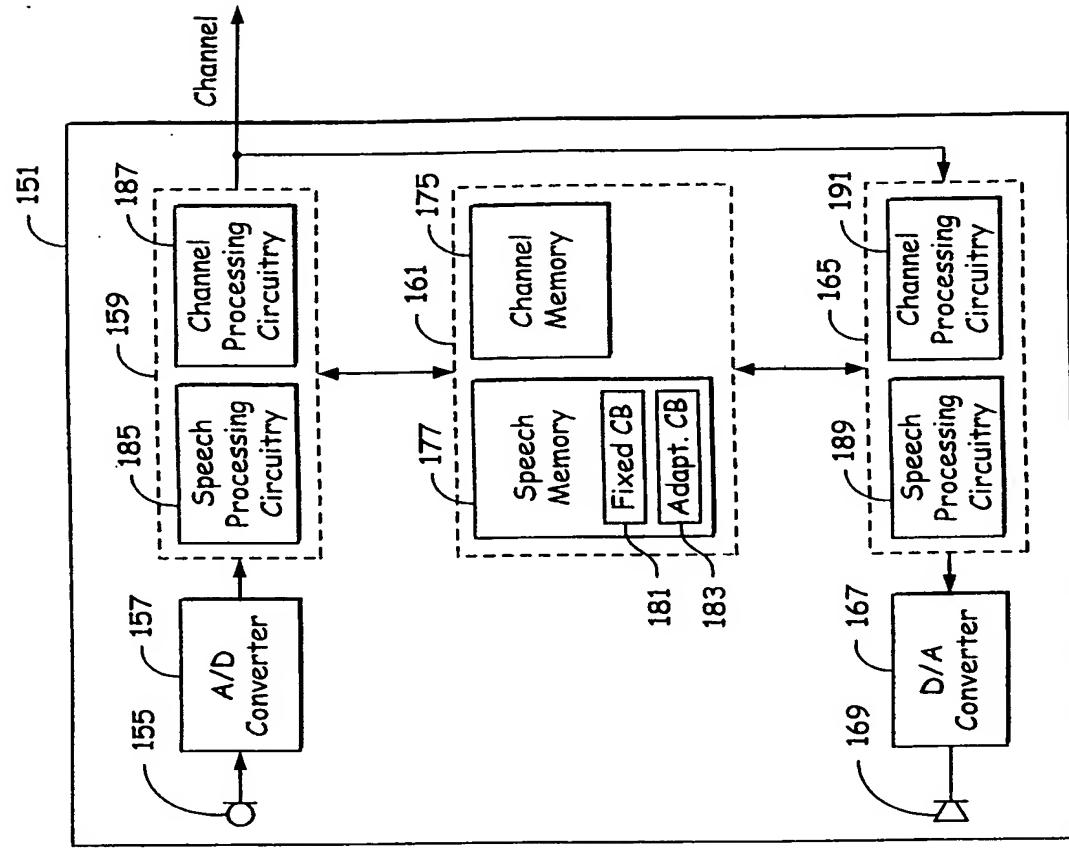
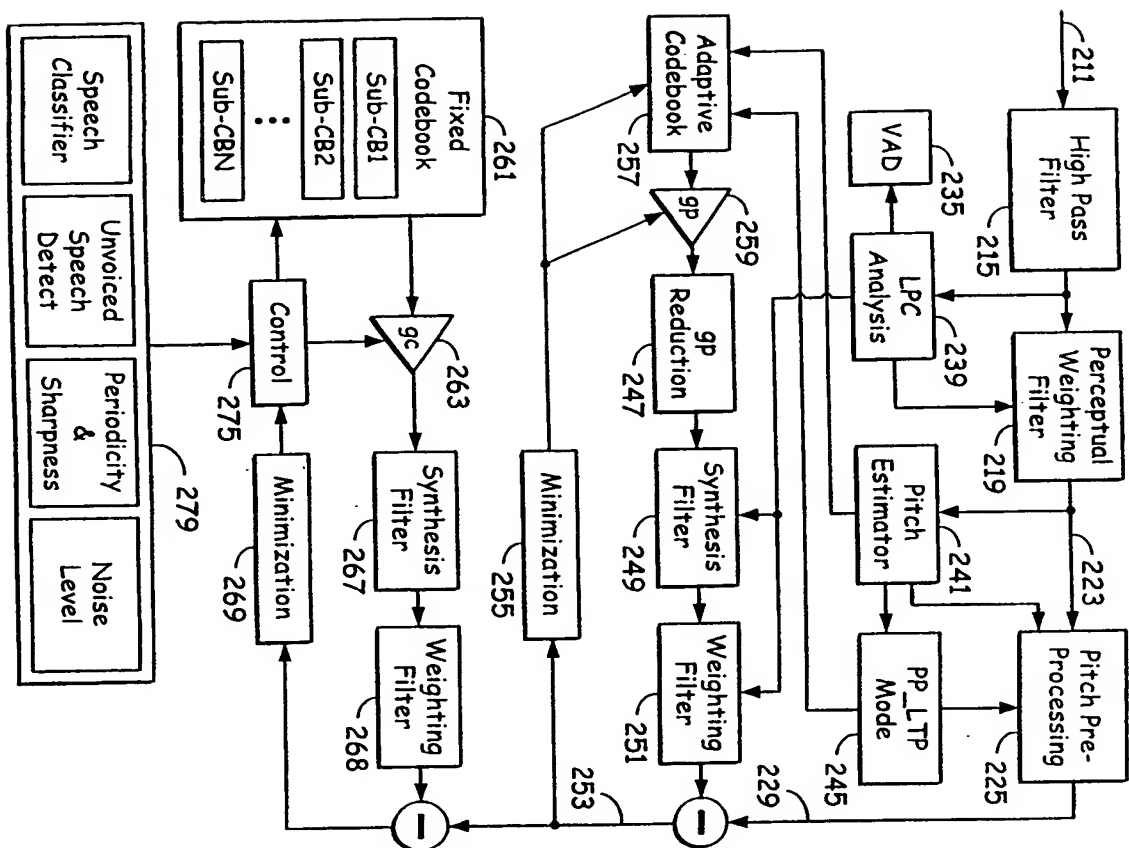
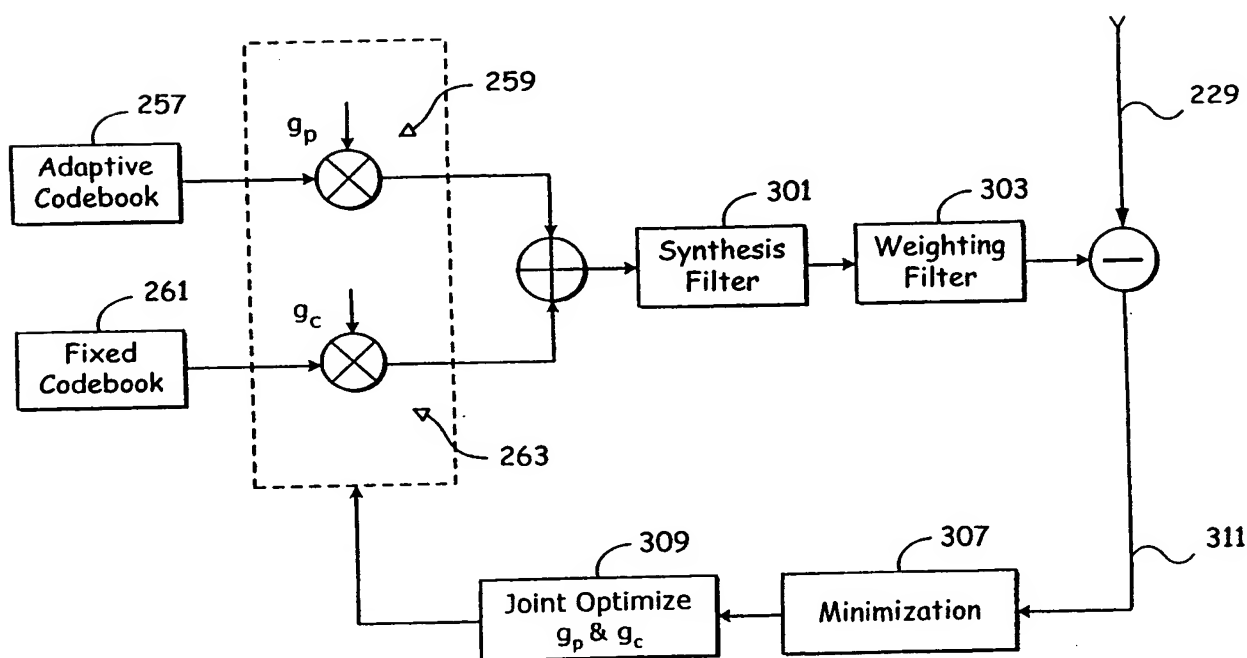


Fig. 1b

3/11



4/11



**Fig. 3**

Fig. 2

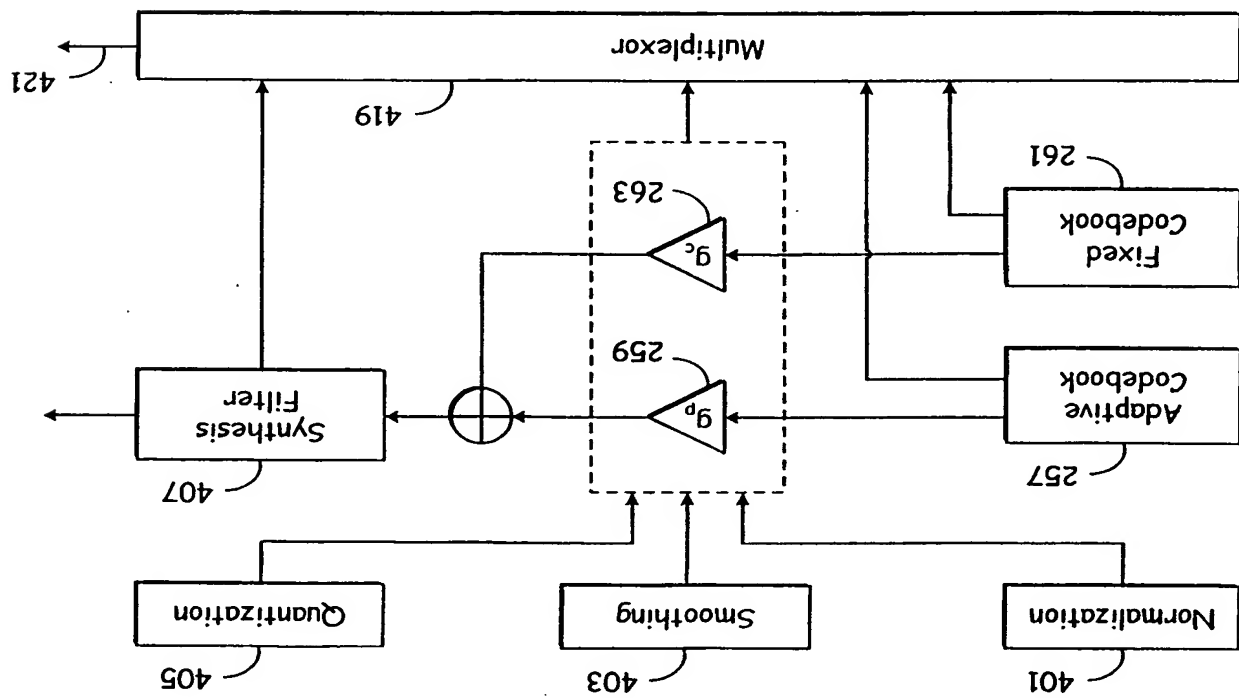


Fig. 4

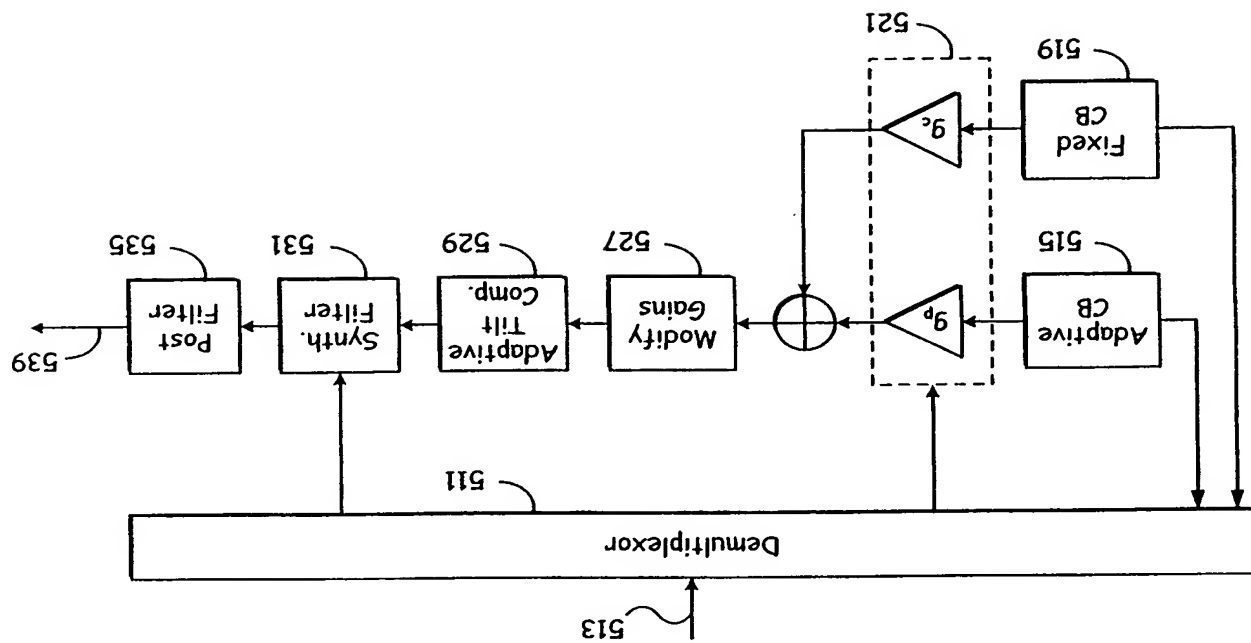
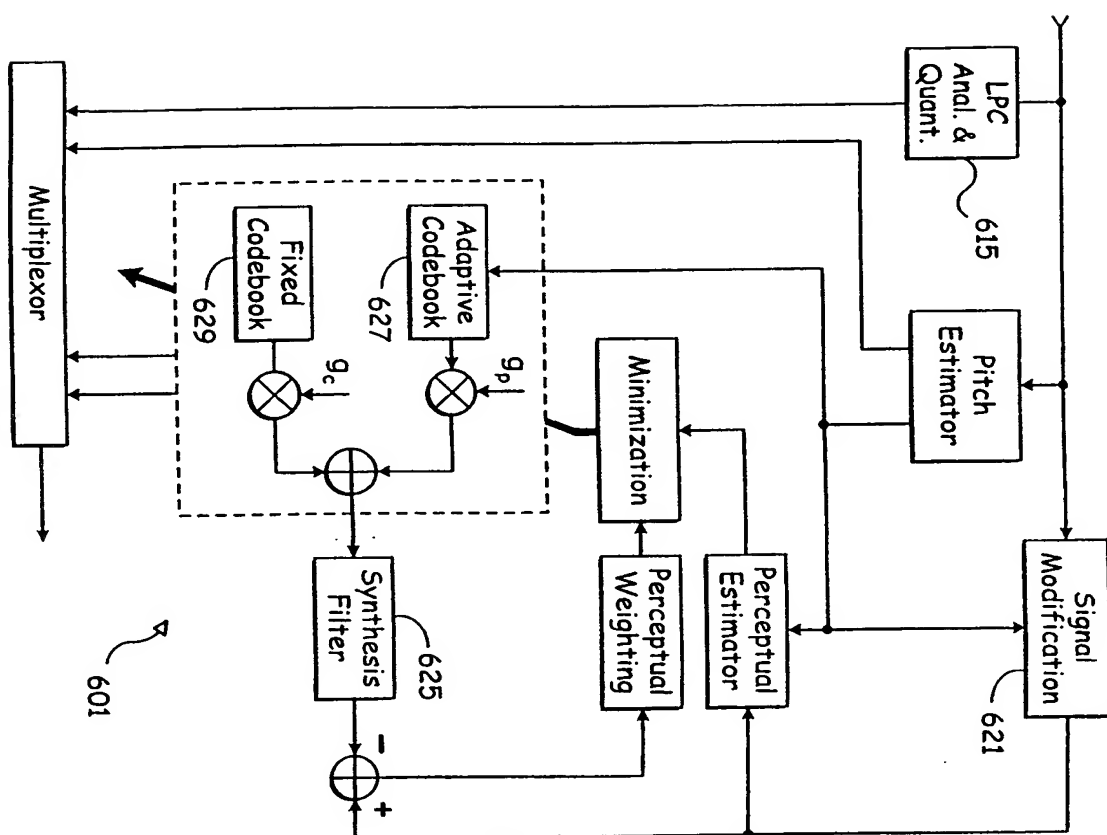


Fig. 5

7/11



8/11

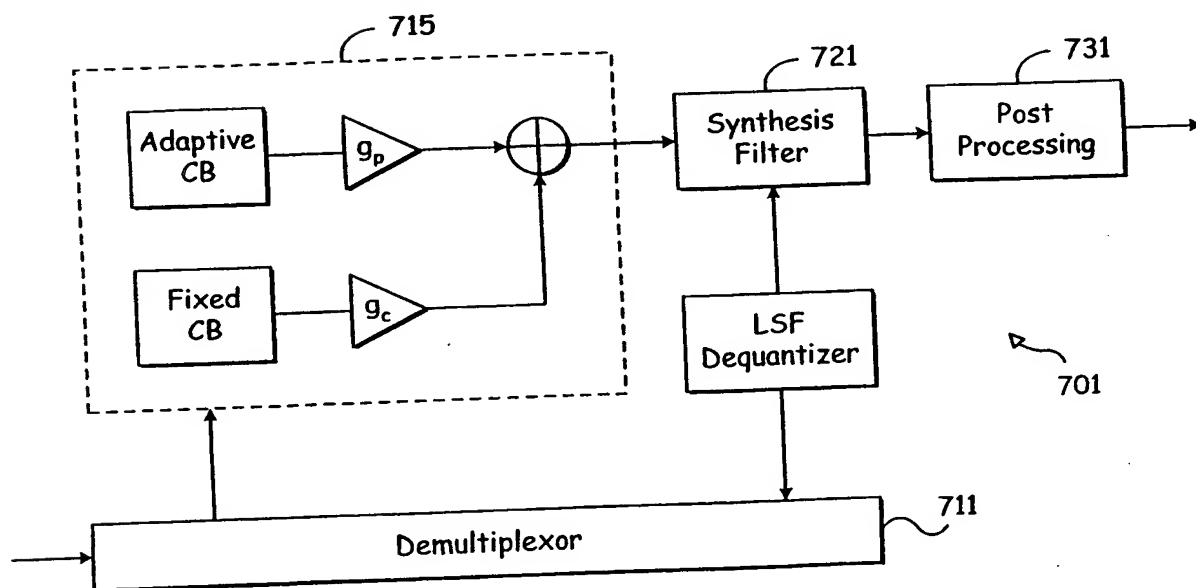


Fig. 7

9/11

10/11

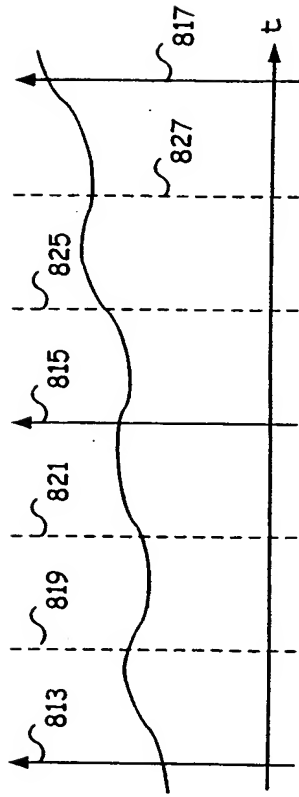


Fig. 8a

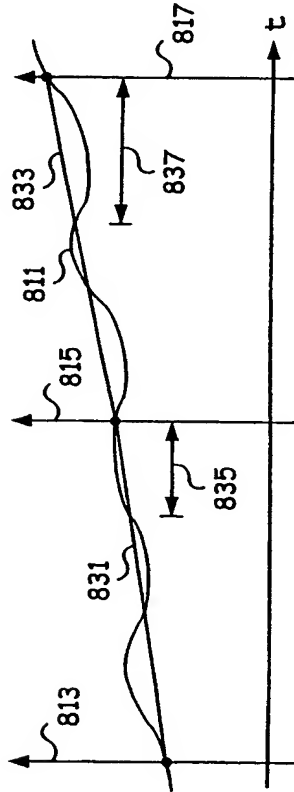


Fig. 8b

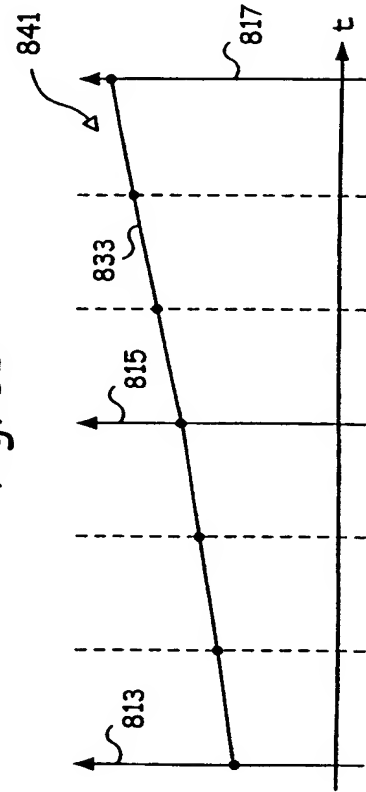


Fig. 8c

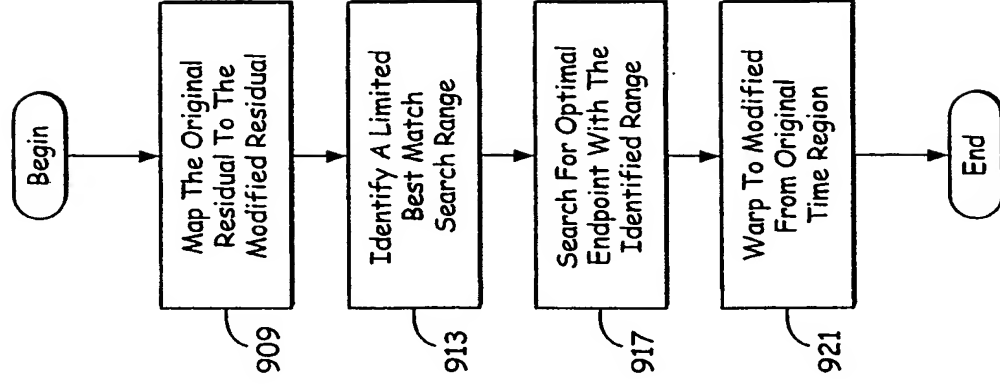


Fig. 9

11/11

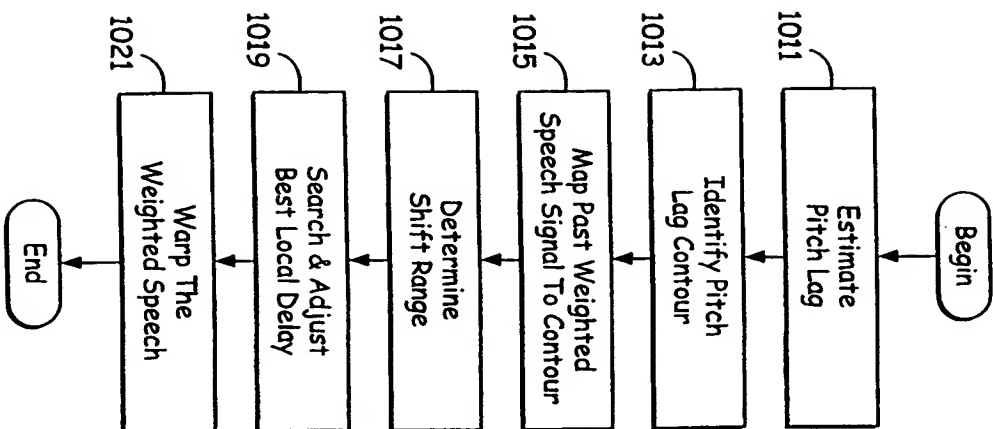


Fig. 10

## INTERNATIONAL SEARCH REPORT

Inventor: Ramos Sánchez, U  
PCT/US 99/19175A. CLASSIFICATION OF SUBJECT MATTER  
IPC 7 610L19/08 610L19/12B. FIELD(S) SEARCHED  
Maximum documentation searched (classification system followed by classification symbols)  
IPC 7 610L

Documentation searched other than maximum documentation to the extent that such documents are included in the fields searched

Electronic data bases consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	KLEIJN M B ET AL: "INTERPOLATION OF THE PITCH-PREDICTOR PARAMETERS IN ANALYSIS-BY-SYNTHESIS SPEECH CODERS" IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, US, IEEE INC, NEW YORK, vol. 2, no. 1, PART 1, page 42-54 XP000423486 ISSN: 1063-6676 page 46 -page 48	1-9, 13
A	ROUAT J ET AL: "A pitch determination and voiced/unvoiced decision algorithm for noisy speech" SPEECH COMMUNICATION, NL, ELSEVIER SCIENCE PUBLISHERS, AMSTERDAM, vol. 21, no. 3, page 191-207 XP004059542 ISSN: 0167-6393 page 194	1, 6, 10

☐ Further documents are listed in the continuation of box C.

☐ Patent family members are listed in annex.

## \* Special categories of cited documents:

- "a" document defining the general state of the art which is not considered to be of particular relevance  
 "b" earlier document but published on or after the international filing date  
 "c" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another document or other special reason (as specified)  
 "d" document relating to an oral disclosure, use, exhibition or other means  
 "e" document published prior to the international filing date but later than the priority date claimed  
 "f" document published after the international filing date

Date of the actual occupation of the international search

10 December 1999

Date of mailing of the international search report

11/01/2000

Name and mailing address of the ISA  
European Patent Office, P.O. Box 18  
NL - 2200 HV Rijswijk  
Tel. (+31-70) 540-2040, Tx. 31 051 epo nl,  
Fax: (+31-70) 540-5010Authorized officer  
Ramos Sánchez, U



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☒ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**